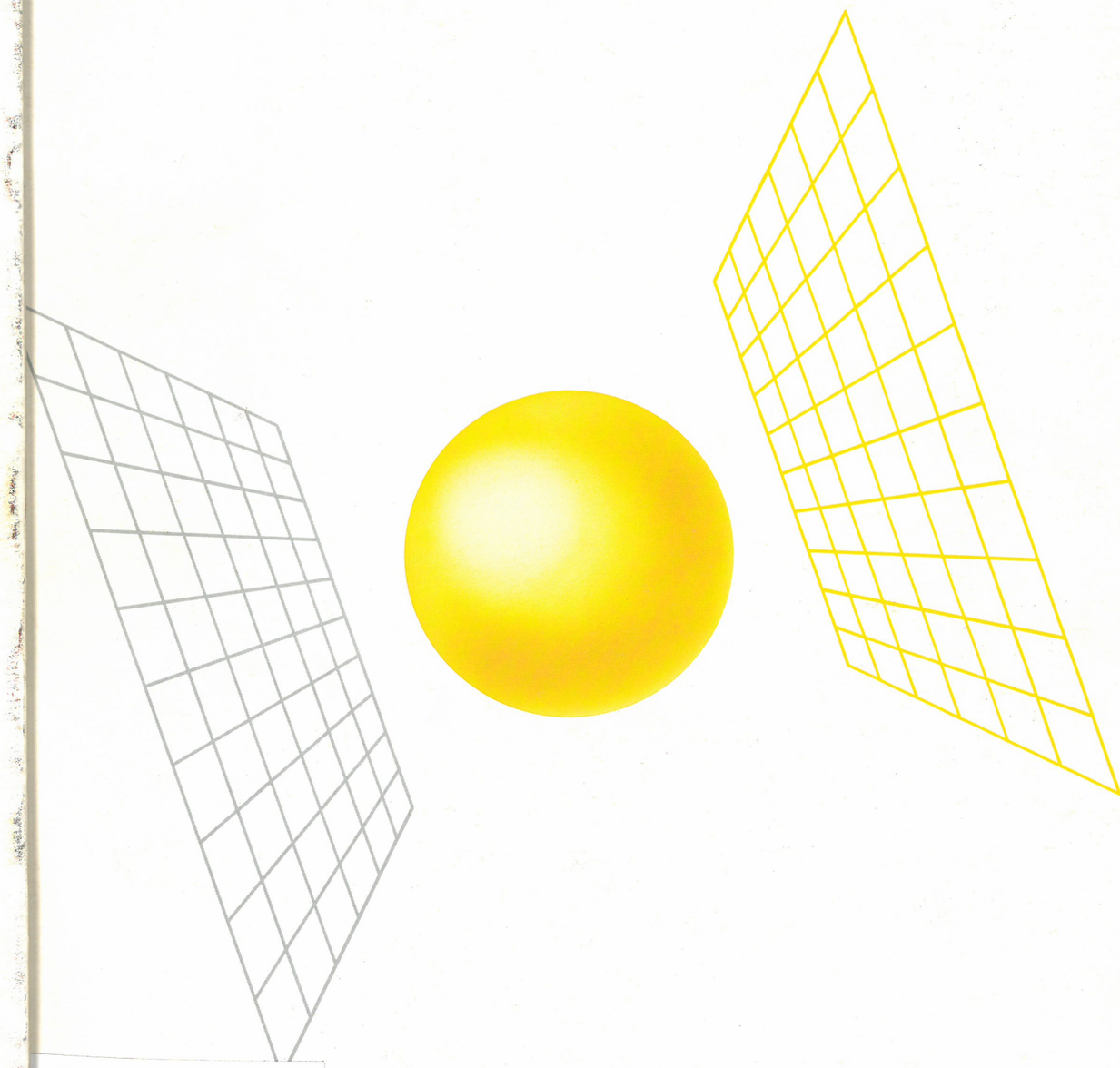
Spedizione in abbonamento postale a tariffa intera/Milano

Quaderni di informatica 25

Rassegna di tecnologia ed applicazioni degli elaboratori elettronici – Anno XI – Numero 2 – 1985



CENTRO STUDI INFORMATICA E AUTOMAZIONE
Honeywell
Honeywell Information Systems Italia

FPDM-70

Honeywell

Honeywell Information Systems Italia

Quaderni di informatica

Rassegna di tecnologia ed applicazioni degli elaboratori elettronici

Anno XI - Numero 2 - 1985

Sommario

Direttore Responsabile
F. Filippazzi

Comitato di Redazione
A. Cicu, C. Falcetti, P. Lupo,
M. Nobile, G. Occhini
F. Sala

Redazione
Honeywell Information Systems Italia
Centro Studi Informatica e Automazione
Via G.M. Vida, 11 - 20127 Milano

Stampa
tipolito Maggioni s.a.s.

Autorizzazione del Tribunale di Milano
n. 93 del 20 Marzo 1974

UN CENTRO PER LA CULTURA INFORMATICA

Con questo numero, la rivista modifica la sua veste grafica: cambia infatti il motivo geometrico che ha caratterizzato la sua copertina per oltre dieci anni. È un cambiamento esteriore che vuole però segnalare un fatto concreto: la costituzione del "Centro Studi Informatica e Automazione", una iniziativa culturale di ampio respiro, nel cui ambito si colloca anche la rivista e la collana di volumi che da essa prende nome.

Obiettivo del Centro è esaminare e interpretare l'evoluzione del settore informatica/automazione per quanto attiene gli aspetti tecnico-scientifici, economico-sociali e di mercato. A tale fine, il Centro ha un Comitato di esperti, la cui composizione rispecchia l'arco delle molteplici tematiche.

Il Centro curerà l'emissione di documentazione articolata su vari livelli, sia assumendo il coordinamento di pubblicazioni già esistenti (come è il caso dei "Quaderni di Informatica"), che creandone di nuove. Oltre a ciò, il Centro promuoverà convegni, seminari e giornate di studio su temi avanzati di tecnologia e applicazioni del settore, con la partecipazione di esperti a livello nazionale ed internazionale.

La Redazione

Come vedono i robot

JAMES E. ORROCK

Honeywell Inc.
Technology Strategy Center
Minneapolis (USA)

1. Introduzione

Si ritiene comunemente che i robot siano in grado di eseguire qualsiasi operazione di fabbricazione ma, in effetti, l'80% dei robot attualmente in uso si limita a compiti quali il caricamento o lo scaricamento di altre macchine, la saldatura a punti e la verniciatura a spruzzo. La maggior parte delle operazioni di fabbrica, in particolare il montaggio, rimangono fuori dalle loro capacità. Il problema risiede, in parte, nel fatto che i "sensi" dei robot sono inadeguati. Ciò è dovuto sia alla immaturità di alcune tecnologie sensoriali, quali i dispositivi di visione e di tatto, sia al fatto che gli algoritmi che interpretano le informazioni fornite dai sensori falliscono se il problema è "non strutturato", ossia se le condizioni sono sostanzialmente diverse da quelle previste. Per sempio, un algoritmo per la identificazione di parti, può fallire se al robot viene presentato un pezzo "sconosciuto" o se l'orientamento di un pezzo "conosciuto" da luogo ad una apparenza "sconosciuta". Oltre a ciò, la maggior parte dei robot sono dotati di rigide strategie di ricerca e manipolazione, che possono facilmente essere messe fuori gioco. Per esempio, un robot può essere incapace di afferrare un pezzo se altri pezzi si trovano lungo il percorso del dispositivo di presa. In definitiva, un robot per il montaggio di parti con dimensioni e forme variabili, collocate in modo casuale, è generalmente troppo lento e soggetto ad errori per essere competitivo con operatori umani.

Negli ultimi 15 anni vi sono stati molti progetti rivolti a sviluppare sensori di visione e di tatto per robot. Tuttavia, questi sforzi sono stati generalmente non coordinati e conseguentemente sono emersi piuttosto lentamente dei dispositivi che integrassero questi due sensi e avessero algoritmi di controllo di alto livello, tali da consentire ai robot strategie alternative di ricerca e manipolazione. Nel 1982, il Technology Strategy Center della Honeywell vinse la gara per il programma "Integrated Vision/Range Sensor Development", finanziato dalla DARPA (Defense Advanced Research Project Agency). L'obiettivo era di sviluppare un sistema sensoriale che consentisse ai robot l'esecuzione di operazioni non strutturate di montaggio e di ispezione.

Nel definire il programma, ritenemmo che il primo passo ragionevole fosse quello di combinare la visione, un senso che ha ricevuto di recente molta attenzione, col "range-finding" (determinazione della distanza), una capacità che è stata invece piuttosto trascurata. Un buon candidato per questo secondo compito era il TCL (Through the Camera Lens), un sistema elettronico di autofuoco per macchine fotografiche reflex, sviluppato di recente dalla Honeywell Visitronic.

Poichè la visione dei robot è comparativamente ben sviluppata, la nostra attività venne concentrata sulla misura della distanza e sulla integrazione di questa con la visione. La percezione della distanza consente al robot di distinguere oggetti che sembrano uguali in una immagine bi-dimensionale, ad esempio rondelle dello stesso diametro ma di spessore differente. Oltre a ciò, le misure di distanza consentono alle dita del manipolatore di afferrare un oggetto senza urtarlo.

Cercare di eguagliare la coordinazione tra mano ed occhio dell'uomo mediante un sistema hardware (sensori) e software (algoritmi) non è impresa banale come potrebbe sembrare dall'elenco di alcuni dei problemi che abbiamo dovuto superare. Per esempio, un sensore che possa essere montato su un "polso" del robot sarebbe più vantaggioso di uno montato su un supporto fissato al piano di lavoro; ma un sensore con ottica mobile peserebbe abbastanza da impedire al braccio del robot di sollevare oggetti pesanti. Pertanto, uno dei quesiti cui dovemmo rispondere fu se il circuito integrato su cui è basato il sistema TCL funziona adeguatamente se la distanza focale è fissa. La risposta fu che ciò dipendeva dall'algoritmo di range-finding. Furono perciò sviluppati e provati diversi algoritmi per determinare quale fosse la soluzione più accurata nel campo d'azione del robot. Gli esperimenti effettuati mostrarono che il sensore veniva ingannato da effetto di parallasse, dalle ombre e dal suo stesso sistema di controllo automatico di amplificazione. Le prove indicarono comunque le strategie da seguire per minimizzare o correggere questi problemi.

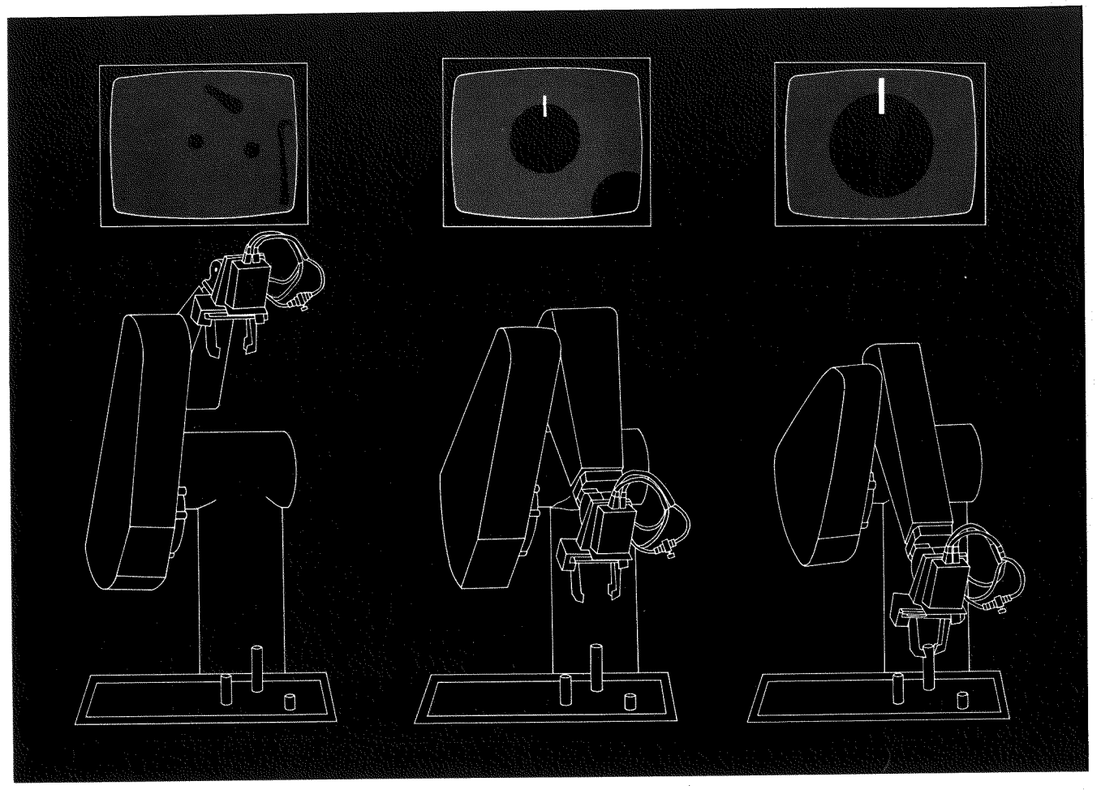


Fig. 1 - Questa sequenza mostra come un robot equipaggiato col sistema di visione bi-dimensionale descritto nell'articolo riesce a scegliere il più alto tra diversi cilindri. Il processo avviene in tre passi. (a) Il braccio (su cui sono collocati i sensori) si porta anzitutto in una posizione fissa da cui l'intera superficie di lavoro viene inquadrata dal dispositivo di visione. L'immagine della superficie di lavoro viene poi elaborata: un algoritmo identifica le aree dell'immagine corrispondenti ai cilindri, ed un altro trasforma le coordinate dei cilindri nel quadro visivo nelle corrispondenti coordinate del tavolo di lavoro. (b) Il braccio si muove su e giù sopra ogni cilindro, posizionando il campo visivo del sensore di range-finding (tacca verticale osservabile sul monitor) a cavallo del bordo del cilindro. Questo sensore misura la distanza dalla testa del cilindro, mediante un sistema di triangolazione ottica. (c) Una volta determinata la distanza per ciascun cilindro, il braccio si posiziona sul più alto di essi e lo preleva. Durante questa operazione vengono effettuate via via misure della distanza tra il cilindro e il manipolatore per controllare la decelerazione del braccio e l'altezza a cui afferrare il cilindro.

2. Definizione dei requisiti.

Ovviamente l'accuratezza richiesta per il dispositivo di visione di un robot dipende dalle caratteristiche del compito che il robot deve svolgere. Noi abbiamo basato le nostre specifiche di progetto sull'analisi dei requisiti richiesti per l'identificazione e l'acquisizione di piccole parti, assumendo come rappresentative le parti che compongono un microswitch Honeywell. La distanza di lavoro del sensore venne definita dallo spazio entro cui opera il robot, ossia un cubo di circa 100 centimetri di lato. L'accuratezza richiesta venne fissata dalle dimensioni delle parti del microswitch.

Durante l'esplorazione iniziale della piattaforma di lavoro, il sensore deve determinare solamente le distanze approssimative per ciascun oggetto. Quindi, per 100 cm l'accuratezza fu fissata in $\pm$ 5 cm. Per distanze minori si richiede ovviamente una maggior accuratezza. Per una distanza di 10 cm (corrispondente alla distanza tra il sensore e le dita del dispositivo di presa) l'accuratezza richiesta venne fissata in base alla differenza in altezza dei due oggetti più simili tra loro, nonchè dal requisito di misurare l'altezza del pezzo più corto con una accuratezza di almeno il 30% in modo da poterlo afferrare senza urtare la base di appoggio. L'accuratezza richiesta per distanze di 10 cm risulta essere di $\pm$0,03 cm.

3. Selezione dei sensori

Il sensore da noi scelto per la visione è un dispositivo a iniezione di carica usato nella telecamera G.E. TN2200. Più precisamente si tratta di una matrice di celle di silicio, ognuna delle quali costituisce un elemento o "pixel" (= picture element) del quadro; ogni cella raccoglie le cariche elettriche generate dalla incidenza della luce sul materiale. Questa matrice fu scelta perchè dotata di una risoluzione sufficiente per l'applicazione prevista e perchè i circuiti elettrici che controllano l'operazione di lettura delle celle sono separati dal sensore. Questo costituiva un vantaggio, in quanto era nostro intendimento collocare in moduli ausiliari la maggior parte della circuiteria del sensore, in modo da minimizzare il peso che il braccio del robot doveva portare.

Occorreva inoltre definire il sistema di range-finding. I due metodi più comunemente usati in robotica per la determinazione delle distanze sono il tempo di volo e la triangolazione ottica. Sono stati considerati sistemi acustici, a radio-frequenza e a laser che misurano il tempo impiegato da un'onda riflessa. Un esempio di tecnica di triangolazione ottica è costituito da un laser che genera una lamina di luce che cade sulla piattaforma di lavoro, e da una telecamera che vede la piattaforma sotto un angolo fisso. L'altezza degli oggetti disposti sulla piattaforma viene calcolata dallo spostamento laterale della luce che essi provocano.

Anche il sistema TCL calcola le distanze per triangolazione. Il ricevitore in questo caso è costituito da una schiera di 24 paia di rivelatori ad accoppiamento di carica. (Un dispositivo ad accoppiamento di carica differisce da uno a iniezione di carica per il fatto che la carica raccolta in una cella è letta come una tensione anzichè come una corrente). La schiera di rivelatori è installata in una telecamera e, per ciascuna coppia di rivelatori, uno riceve luce proveniente da un settore della lente (luce A) e l'altro dall'altro settore (luce B), come illustrato in fig. 2. La distanza tra i due settori serve da base per la triangolazione. I rivelatori A registrano una "traccia" (variazione della luminosità della scena nella porzione dell'immagine che cade sulla schiera di dispositivi) e quelli B ne registrano un'altra. Le tracce risultano sovrapposte quando la lente di fronte alla schiera è a fuoco sull'oggetto, altrimenti esse sono spostate l'una rispetto all'altra.

Il valore dello spostamento è proporzionale all'errore focale, mentre il segno dello spostamento indica se l'oggetto è dentro o fuori della distanza focale. Il calcolo dello spostamento delle tracce fornisce valore zero se l'oggetto è nel fuoco della lente TCL, un valore negativo se è più vicino alla lente della distanza focale, un valore positivo se è più lontano. La distanza di un oggetto può quindi essere determinata in base alla distanza focale della lente e all'errore focale misurato attraverso lo spostamento delle tracce.

Dopo aver esaminato i vari metodi di range-finding, arrivammo alla conclusione che il sistema TCL era il migliore per la nostra applicazione, in quanto esso è di tipo passivo e poichè consiste di pochi circuiti integrati e quindi è piccolo e relativamente poco costoso.

4. Algoritmo per la elaborazione della visione.

Il sensore TN 2200 prima menzionato venne collegato ad un processor di visione VS-100 della Machine Intelligence Corporation (MIC). Il sistema di visione della MIC include un certo numero di algoritmi per l'elaborazione dell'immagine sviluppati originariamente dallo Stanford Research Institute (SRI). L'utente ha facoltà di scelta tra questi algoritmi e può fissare i parametri di ciascun algoritmo in base alla sua applicazione. Noi abbiamo sviluppato, interattivamente col sistema MIC, due algoritmi diretti, rispettivamente, alla identificazione di parti e alla loro localizzazione.

Il sistema MIC converte l'immagine analogica (scala di grigi) prodotta dalla matrice di sensori in una immagine digitale (bianco e nero). Le aree luminose dell'immagine vengono delimitate, nel senso che sono elaborati soltanto gruppi contigui di pixel o "blobs" (macchie). Successivamente sono calcolate le caratteristiche globali di base per ciascun blob, quali l'area, il centro di gravità e il "peround" (una misura della rotondità). Durante una sessione di addestramento, l'utente sceglie il gruppo di caratteristiche da impiegare per l'identificazione delle parti, e crea una base di dati delle caratteristiche delle parti previste e delle tolleranze relative. Poichè il robot sarebbe stato provato su parti semplici, abbiamo scelto un gruppo ristretto di caratteristiche. Durante il funzionamento, il processor di visione calcola i valori relativi a ciascun blob e li confronta coi valori presenti nella base di dati. Se si trova un appaiamento entro le tolleranze fissate, il blob viene riconosciuto come una delle parti previste.

Un altro algoritmo determina le coordinate x, y di ciascun oggetto dell'immagine. L'immagine della piattaforma di lavoro elaborata dagli algoritmi di identificazione e localizzazione delle parti viene acquisita a partire da un punto fisso dello spazio di lavoro. L'algoritmo di identificazione fornisce le coordinate del centro di gravità delle parti in rapporto al campo visivo del sensore, indicando, ad esempio, che una parte è centrata sul pixel 75.100. Il controllore del robot prende allora nota delle coordinate del punto di presa (punto intermedio tra le dita del manipolatore) nell'area di lavoro.

Dopo aver montato il sensore sul robot, un oggetto venne posto a caso sul tavolo. Il manipolatore venne posizionato dapprima sul punto fisso dell'area di lavoro e successivamente sulla posizione di presa. La trasformazione tra le coordinate del quadro visivo e quelle dello spazio di lavoro viene effettuata confrontando le coordinate del primo, fornite dall'algoritmo di identificazione nella prima posizione, con le coordinate del tavolo di lavoro, fornite dal controllore del robot nella seconda posizione. Durante il funzionamento, le coordinate del quadro visivo relative ad un oggetto vengono sottoposte a questa trasformazione, che fornisce pertanto le coordinate x, y su cui deve essere centrato il manipolatore, sopra l'oggetto. Il sensore venne montato in modo che l'asse del campo visivo del TCL fos-

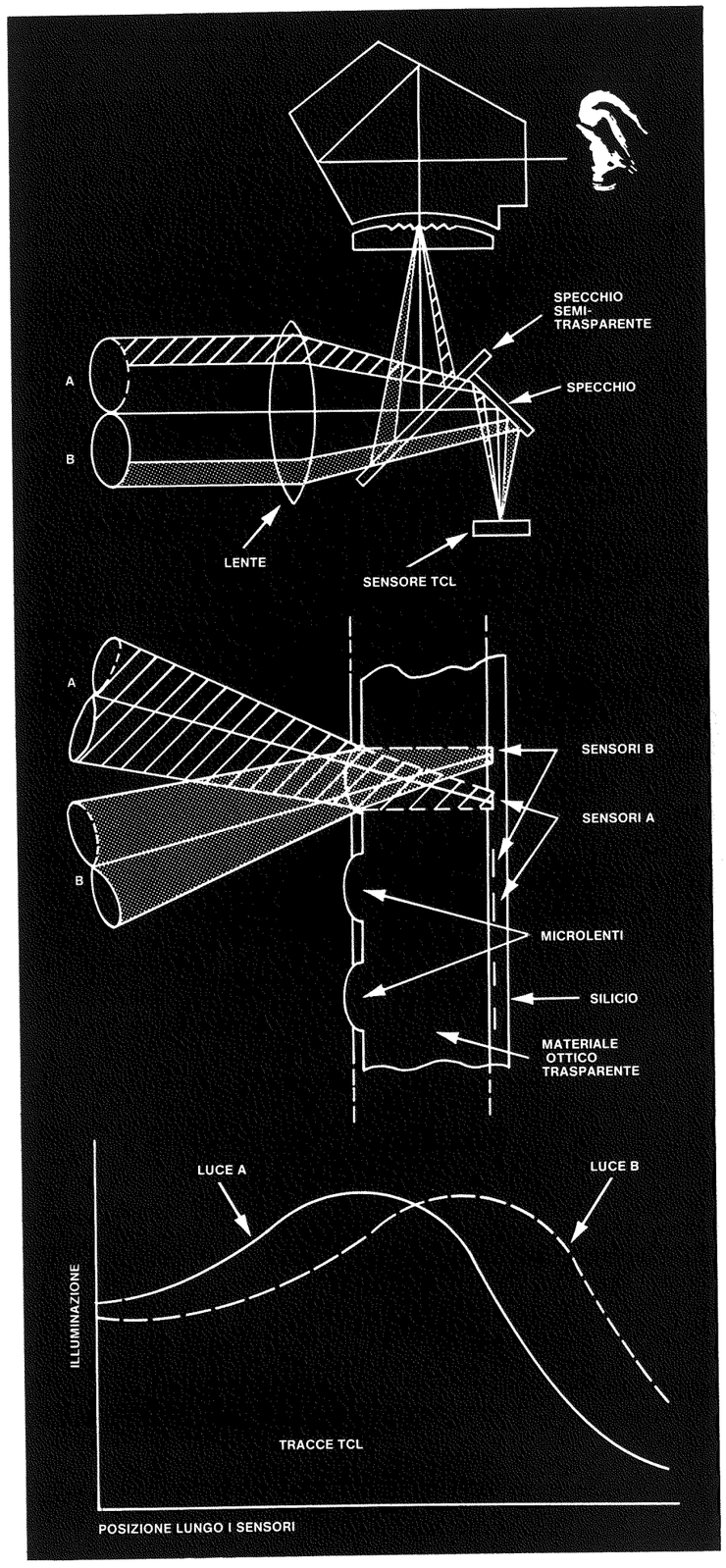


Fig. 2 - Il sensore di range-finding del robot è costituito da un gruppo di chip progettati dalla Honeywell Visitronic come sistema di autofocalizzazione per macchine fotografiche reflex. In questo sistema (denominato TCL) una doppia fila di sensori (A e B) viene colpita da luce proveniente dalla lente della macchina. Le microlenti collocate sopra il chip in cui sono realizzati i sensori, separano la luce proveniente da due differenti settori della lente della macchina, dirigendola su sensori adiacenti. La carica elettrica dei rivelatori A costituisce una "traccia" (variazione della intensità luminosa lungo il segmento dell'immagine che cade sul dispositivo), mentre un'altra "traccia" è fornita dai rivelatori B. Se la lente è a fuoco sull'oggetto, le due tracce risultano sovrapposte; in caso contrario (come nella figura) esse sono separate da una distanza che è proporzionale all'errore focale.

se lo stesso dell'asse lungo il quale si chiudono le dita del manipolatore. Portando quindi il manipolatore in corrispondenza di queste coordinate x,y, il misuratore di distanza risulta allineato sul pezzo.

5. Algoritmo di range-finding

L'accuratezza delle misure di distanza, nell'ambito dello spazio di lavoro previsto, viene fissata dall'algoritmo di range-finding. Lo spostamento delle tracce del TCL può essere misurato in parecchi modi diversi. Alcuni metodi, quali la misura della distanza delle tracce a livelli fissi dell'intensità luminosa, forniscono però risultati che variano col livello della luce, la forma delle tracce e altri fattori estranei. Furono pertanto presi in esame diversi metodi per calcolare lo spostamento delle tracce e furono sperimentati due algoritmi, denominati algoritmo di intensità media e algoritmo H_F, per determinare quale fosse il più accurato.

Come indicato dal nome, il primo algoritmo calcola l'intensità media di un certo gruppo di tracce e misura lo spostamento in corrispondenza di tale valore di ordinata. Poichè la retta che rappresenta la media può intercettare ciascuna traccia in più di un punto, l'algoritmo include una routine che confronta la pendenza delle curve, per garantire che la misura venga effettuata su porzioni corrispondenti delle tracce A e B.

L'algoritmo H_F, che è stato sviluppato dalla Honeywell Visitronic, opera su tutti i punti di una traccia anzichè su un punto solo. L'algoritmo considera la parte di traccia corrispondente ad ogni coppia di pixel, per ciascuna di esse effettua un calcolo il cui effetto è di moltiplicare la pendenza media di ciascun segmento delle tracce per l'area da esse racchiusa, e infine somma i valori ottenuti da tutte le paia di pixel. Tale somma, detta H_F, è zero se le tracce coincidono, in quanto l'area è nulla. Se si ha invece un valore non nullo, l'algoritmo cerca di minimizzare H_F spostando le tracce, un pixel alla volta, e ricalcolando H_F dopo ogni spostamento. H_F cambia segno quando le tracce si incrociano. A questo punto si ottiene, con la precisione di un pixel, il numero di pixel di cui occorre spostarsi per azzerare H_F. La frazione di pixel che deve essere aggiunta al numero intero per rendere H_F esattamente nullo può essere determinata per interpolazione. Questo algoritmo è illustrato in fig. 3.

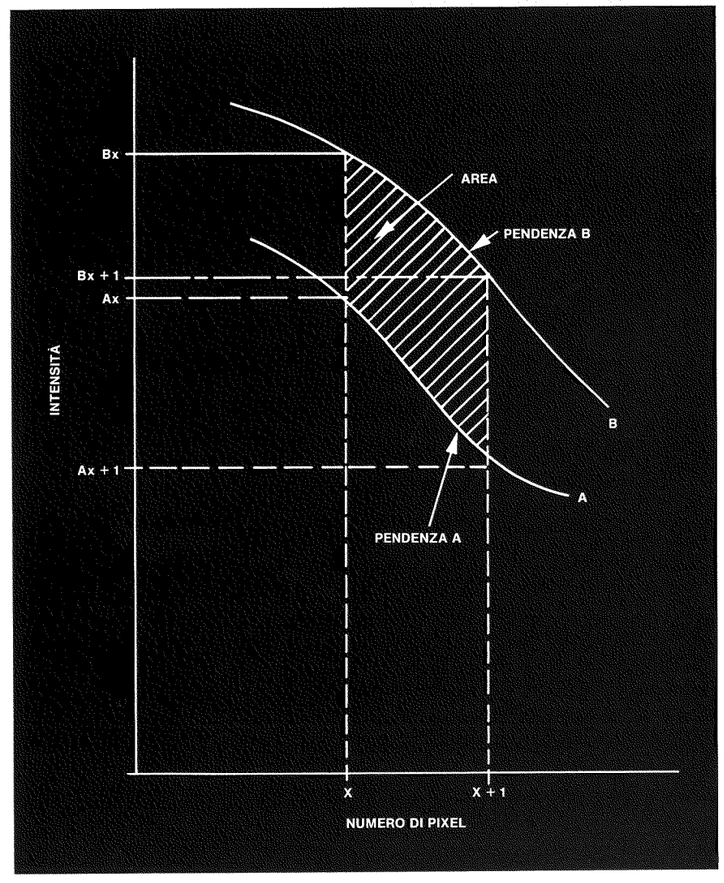


Fig. 3 - L'algoritmo di range-finding calcola la distanza tra le due "tracce" spostando una di esse, pixel dopo pixel, fino a sovrapporla all'altra. L'algoritmo calcola una funzione proporzionale all'area tra le due tracce. Se il valore della funzione è zero, le due curve sono sovrapposte; se non è zero, l'algoritmo sposta di un pixel una delle curve e ripete il calcolo. Quando la funzione cambia segno, significa che le due curve si sono incrociate. L'algoritmo calcola allora, per interpolazione, la frazione di pixel che deve essere aggiunta per ottenere la sovrapposizione perfetta.

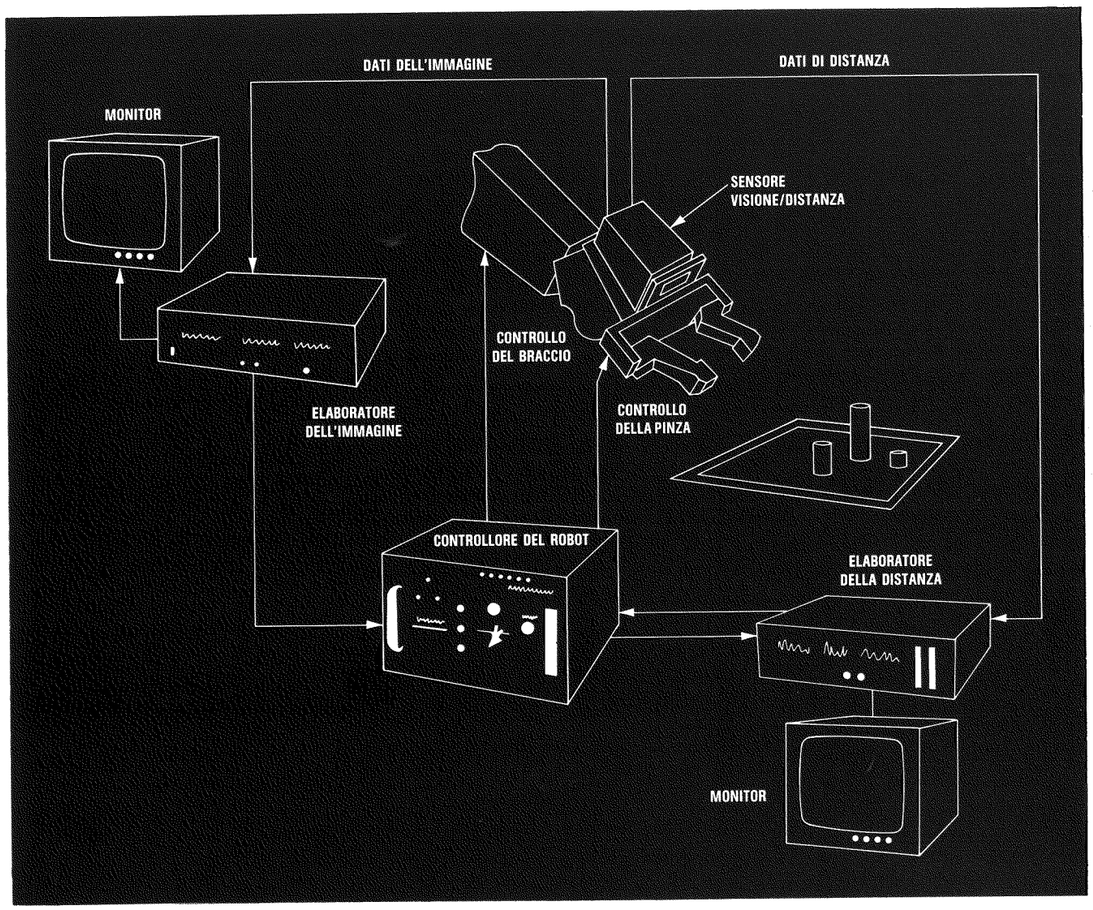


Fig. 4 - L'apparato sperimentale comprende un monitor, connesso all'elaboratore di immagine, su cui viene presentata l'uscita del sensore di visione nonchè la tacca del sensore di distanza. Su un altro monitor sono invece visualizzate le due tracce del sistema TCL per il rilevamento della distanza. L'uso integrato dei due monitor è stato molto utile per la comprensione dei problemi emersi durante le prove.

6. Il banco ottico

Le prime verifiche sul sensore vennero effettuate con un banco ottico e un oggetto di calibrazione piatto. Vennero tracciati gli spostamenti (in pixel) delle tracce, calcolati dal sensore nella fascia di distanze da 10 a 100 cm. Venne poi cambiata la distanza focale delle lente del sensore e ripetute le misure. Furono provati entrambi gli algoritmi di range-finding. Il confronto delle due famiglie di curve mostrò che l'algoritmo H_F era più accurato, soprattutto perchè la routine di confronto della pendenza nell'altro algoritmo falliva quando lo spostamento delle tracce era elevato. La migliore distanza focale venne scelta impiegando una curva molto ripida nella famiglia H_F. Infatti, quanto maggiore è la pendenza della curva, tanto maggiore è la differenza tra gli spostamenti corrispondenti a due differenti distanze e quindi tanto più accurata risulta la misura. Vennero poi determinati la ripetitività dei valori di spostamento e l'accuratezza delle misure di distanza (data dalla distanza focale della lente e dall'errore focale corrispondente allo spostamento calcolato).

Per fare in modo che il sensore potesse fornire direttamente la distanza, fu creata una tabella che correlava lo spostamento delle tracce alla distanza, spostando l'oggetto verso il sensore con incrementi di un centimetro e registrando ad ogni passo lo spostamento (in pixel) calcolato dall'algoritmo H_F. L'algoritmo finale calcolava lo spostamento e derivava il corrispondente valore della distanza dalla tabella precedente, oppure, nel caso che lo spostamento calcolato non trovasse riscontro nella tabella, l'algoritmo determinava la distanza per interpolazione dei valori registrati. Fu determinata l'accuratezza dell'intero algoritmo: essa risultò essere di circa $\pm$ 3 cm a 100 cm e $\pm$ 0,03 cm a 10 cm, cioè entro le specifiche poste.

7. Il test dei cilindri

Il passo successivo fu la realizzazione di un sistema di dimostrazione, con un sensore montato su un robot PUMA 560, e la verifica della capacità del sensore di guidare il robot nel duplice compito di identificazione e prelievo delle parti. Era in effetti concepibile che l'accuratezza della misura della distanza venisse influenzata da urti meccanici, disturbi elettrici, variazioni di illuminazione, presenza di ombre o imprecisione delle informazioni fornite al sistema.

I termini della dimostrazione vennero definiti in base all'obiettivo del progetto, che era di progettare un possibile sensore piuttosto che un intero sistema. Conseguentemente, fu trascurato il problema di determinare l'orientamento di parti asimmetriche ed il sensore venne sperimentato usando cilindri e rondelle.

I pezzi vennero presentati al robot su un piatto; essi erano tra loro ben separati. Infine non venne preso in considerazione il montaggio dei pezzi.

Nel primo ciclo di esperimenti, vennero messi a caso sul tavolo, sotto il robot, dei cilindri di eguale diametro e di altezza variante da 2,5 cm a 12,7 cm. Il braccio del robot pose il sensore ad un punto fisso del tavolo e le coordinate x, y di ciascun cilindro vennero determinate mediante l'algoritmo di elaborazione della visione. Il sensore venne poi posizionato circa 30 cm al di sopra di ciascun cilindro, col campo visivo del dispositivo TCL a cavallo del bordo del cilindro. Vennero così misurate le distanze dai cilindri.

Una volta identificato il più alto dei cilindri confrontando le misure di distanza, il braccio del robot si abbassò per prelevarlo. Durante questo movimento, la distanza venne misurata in continuazione. La velocità del braccio rimase costante finchè la distanza non raggiunse un valore di soglia prefissato, scelto in modo da portare il braccio a fermarsi alla corretta altezza di presa, dopo una decelerazione graduale. Il robot afferrò allora il cilindro e lo pose su un lato del tavolo di lavoro.

Durante la prova, vennero tenuti sotto osservazione il robot, l'immagine della scena che, insieme con la posizione del campo visivo del TCL, veniva rappresentata su un monitor collegato col processor di visione MIC, e le tracce TCL che erano rappresentate su un monitor connesso col processore di distanza. Tre problemi emersero da questo esame. Anzitutto, il posizionamento del campo visivo del TCL sul bordo dei cilindri variava sensibilmente da cilindro a cilindro. In secondo luogo, malgrado l'accuratezza delle misure statiche di distanza, il sensore aveva difficoltà a distinguere cilindri aventi una differenza in altezza inferiore a 1,5 cm. Infine, il posizionamento delle dita del robot sui cilindri variava da cilindro a cilindro. Per determinare quali fossero le cause di questi problemi, dividemmo il processo in vari passi ed esaminammo ciascuno di essi in dettaglio.

8. Effetto di parallasse

La variabilità nel posizionamento del sensore TCL era in parte dovuta

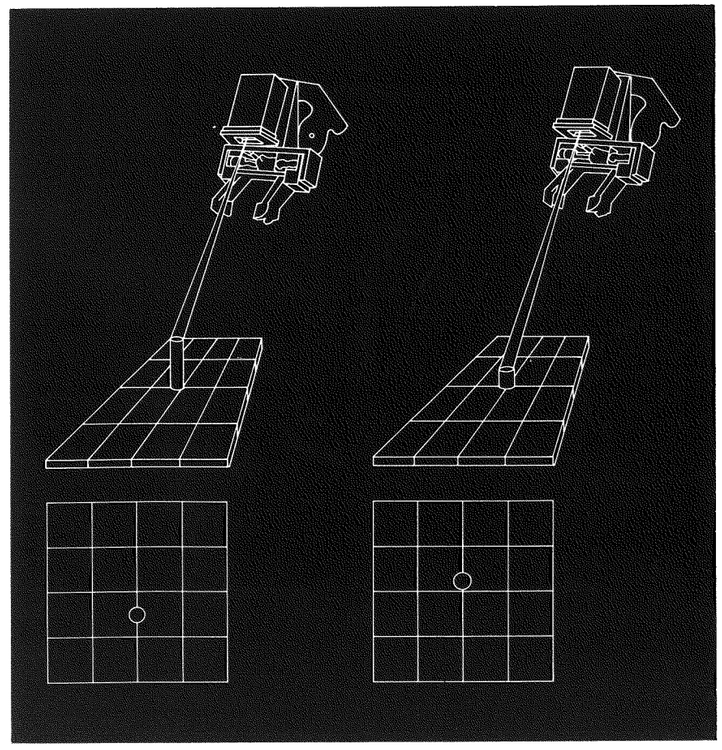


Fig. 5 - La parallasse è causa di errori di posizionamento. Le figure illustrano come il sistema fornisca posizioni diverse a seconda l'altezza dell'oggetto. Il problema si può ovviare per approssimazioni successive, con una ulteriore misura, eseguita nel punto determinato in precedenza.

a parallasse. Infatti, per il sensore, la posizione di un cilindro rispetto alla superficie di lavoro sembra cambiare quando varia l'altezza del cilindro (v. fig.5). Nelle prove eseguite per verificare la trasformazione di coordinate tra il campo del sensore di visione e le dita del robot, era stato usato un cilindro alto 7,5 cm. Quando questo venne sostituito con uno più alto, alla base superiore del cilindro vennero assegnate delle coordinate x, y che erano più lontane di quelle effettive. Con un cilindro più corto, la posizione apparente risultò invece più vicina di quella reale. Questo problema poteva però essere risolto ricalcolando la posizione del cilindro dopo che il manipolatore si era posto nella posizione x, y approssimata, calcolata precedentemente.

9. Effetto delle ombre

Poichè la determinazione della posizione di un cilindro risultava talora erronea, venne cambiata la posizione del campo visivo di TCL rispetto al cilindro, e si constatò che ciò dava luogo a valori diversi dello spostamento calcolato delle tracce.

Si attribuì questo inconveniente all'effetto delle ombre, per cui l'algoritmo H_F calcolava in effetti una distanza che era la media della distanza dall'ombra e di quella dal cilindro. Non è possibile in pratica, con nessun sistema di illuminazione, eliminare completamente le ombre e in ogni caso il problema si ripresenta quando si ha a che fare con piccoli pezzi molto vicini tra loro. Una soluzione migliore era di trovare il modo di selezionare opportunamente una porzione delle tracce da cui calcolare la distanza.

Poichè la variazione dell'intensità luminosa dentro un'ombra è graduale rispetto a quella che si ha attraversando il bordo di un cilindro, la soluzione ovvia era di fissare una soglia per la pendenza delle tracce e di calcolare lo spostamento usando solo quelle porzioni di traccia aventi pendenza superiore alla soglia. Venne perciò realizzata una routine che segmentava la porzione di traccia corrispondente al bordo del cilindro. Con ciò, il calcolo dello spostamento delle tracce divenne assai più affidabile.

10. Effetto della forma delle tracce

Le ombre possono influenzare la capacità del robot di distinguere parti di altezze poco diverse tra loro ma non dovrebbero influire sulla scelta dell'altezza a cui afferrarli, in quanto il cilindro riempie all'incirca il campo visivo durante l'ultima misura di distanza prima della presa ed essendo minima la variazione di intensità dentro l'ombra nelle immediate vicinanze del cilindro. Tuttavia, mentre il robot afferrava i cilindri, venne notato che la forma delle tracce variava da tentativo a tentativo. Più precisamente, variava la pendenza della porzione corrispondente al bordo del cilindro e tale variazione sembrava influire sul punto scelto dal robot per la presa.

La variabilità della pendenza non poteva essere attribuita alle ombre in quanto in questo caso il sensore puntava sul lato illuminato dei cilindri. La posizione del campo visivo del sensore di distanza variava, ma ciò avrebbe dovuto influire solo sulla posizione del bordo nell'ambito delle tracce, e non sulla ripidità della pendenza in corrispondenza del bordo. Alla fine, il colpevole risultò essere il controllo automatico di amplificazione (AGC = Automatic Gain Control) del sensore di distanza. Allorchè la schiera di elementi del TCL viene interrogata, un comparatore realizzato in un chip associato determina la tensione media sulle porte del rivelatore e la confronta col valore prefissato del AGC. I rivelatori possono raccogliere cariche elettriche finchè la tensione media raggiunge tale valore. Lo scopo di questa caratteristica è di rendere il TCL insensibile al livello di luce, ma risultò che esso rendeva la pendenza delle tracce dipendente dal posizionamento del TCL.

Se il TCL è posizionato primariamente sulle caratteristiche di fondo, essendo la luminosità del cilindro assai maggiore di quella media, le tracce presentano una pendenza elevata in corrispondenza del bordo del cilindro. Se invece il campo visivo del TCL è posizionato primariamente sul cilindro, la luminosità di quest'ultimo non risulta molto diversa da quella media e la pendenza attraverso il bordo del cilindro risulta meno ripida. Entrambe le tracce A e B hanno la stessa pendenza, per cui non viene influenzata la determinazione del segno durante il calcolo dello spostamento; ma l'interpolazione finale assumeva una pendenza invariante, per cui le variazioni di pendenza riscontrate potevano causare variazioni apprezzabili della distanza calcolata.

Per risolvere questo problema sono state studiate tecniche di normalizzazione della forma delle tracce. Si è trovato che era possibile rendere le tracce simili, a prescindere dalla posizione del campo visivo del TCL, tarando manualmente lo AGC, col risultato di rendere il valore calcolato dello spostamento praticamente invariante. Ciò suggerisce che il problema del posizionamento potrebbe essere tollerato sostituendo lo AGC con un algoritmo che misuri una caratteristica delle tracce, quale l'ampiezza di picco, o che confronti le tracce con delle sagome opportune.

11. Ulteriori test

Dopo gli esperimenti coi cilindri, venne effettuata una serie di prove con piccole rondelle. Due rondelle che differivano tra loro solo nello spessore furono messe a caso sul tavolo di lavoro. Il robot le localizzò mediante il sensore di visione, misurò la distanza di ciascuna per determinare quale delle due fosse più spessa e afferrò e spostò quest'ultima. L'esperimento mostrò che i problemi sorti quando il robot lavorava con cilindri alti non si avevano lavorando con parti sottili. In particolare, risultava assai minore la variazione di posizionamento del TCL sul bordo dell'oggetto. Ciò era dovuto a diverse ragioni. Anzitutto, la parallasse è minore quando i pezzi sono bassi e le ombre possono essere controllate più facilmente. Inoltre, essendo gli oggetti bassi, il sensore lavorava su distanze minori; si ottenevano pertanto immagini con definizione più elevata, che rendeva più accurata la determinazione della posizione degli oggetti. L'accuratezza nella determinazione delle coordi-

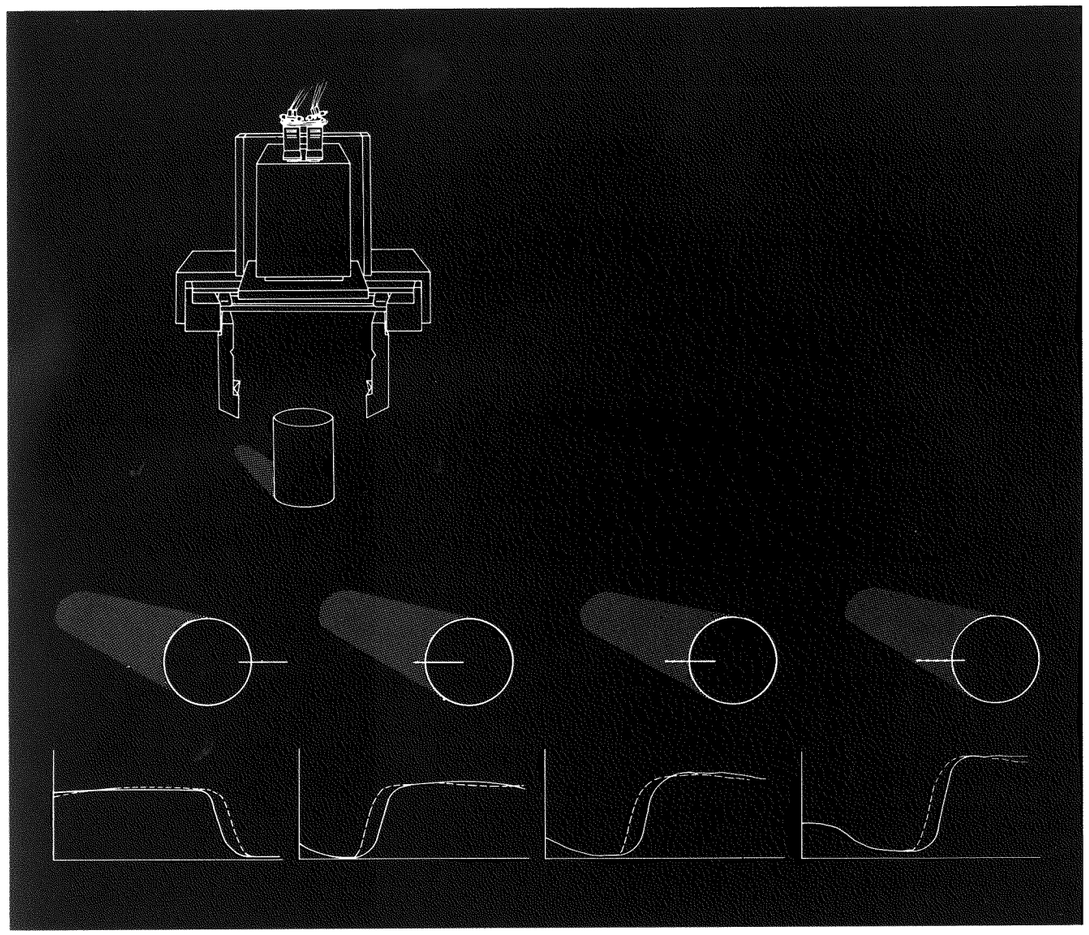


Fig. 6 - Le ombre possono causare errori nella determinazione della distanza, per il fatto che quest'ultima viene calcolata spostando le due curve TCL fino a sovrapporle. Il problema è stato superato effettuando tale operazione solo su opportuni tratti delle due curve.

nate x, y riduceva, a sua volta, l'errore nel posizionamento del campo visivo del TCL sulla rondella. L'invarianza nel posizionamento del TCL aveva l'effetto di minimizzare i problemi causati dalle ombre e dal sistema di AGC. In sostanza, poichè tutti i problemi sono tra loro interconnessi, il miglioramento in un'area portava al miglioramento anche nelle altre.

12. Conclusioni

Gli esperimenti coi cilindri hanno messo in luce problemi indicanti la necessità di algoritmi di elaborazione più raffinati. I test con le rondelle hanno mostrato però che gli algoritmi esistenti sono adeguati se la variazione in altezza delle parti non è estrema. A distanze ravvicinate (10 - 15 cm) il robot è capace di distinguere in modo sicuro pezzi con altezze che differiscono per non più di 0,5 mm. Questo risultato è paragonabile o migliore rispetto a quelli ottenuti con altri sistemi di visione.

13. Ringraziamenti

Il team del Technology Strategy Center che ha lavorato sul sensore di visione era costituito da Ken Dady, Steve Scarborough, Jim Garfunkel, Tom Hill, Chuck Raymond, Basil Owen, oltre che dall'autore di questo articolo. L'algoritmo H_F è stato sviluppato da Norm Stauffer, Denny Wilwerding e Jim Adams della Visitronic, che hanno fornito inoltre un supporto tecnico durante tutto il progetto.

Il materiale di questo articolo è stato tratto da "Scientific Honeyweller", vol. 5, n. 3, settembre 1984.

Reti locali di elaboratori

CORNELL DRENTEA

Honeywell Inc.
Minneapolis (USA)

1. Introduzione

Questo articolo fa il punto sulle reti locali di elaboratori o LAN (Local Area Networks) e più specificamente sull'hardware che consente la comunicazione ad alta velocità tra più computer, e sulle interfacce hardware e software richieste perché la rete permetta di collegare computer tra loro dissimili. Idealmente, una LAN dovrebbe soddisfare quattro requisiti di base. Il primo è un basso costo della rete, in modo che l'onere della comunicazione sia proporzionato al costo attuale dei piccoli elaboratori. Il secondo requisito è che la rete sia basata su un mezzo trasmissivo a larga banda, in modo che la velocità di comunicazione sia limitata dai computer piuttosto che dal mezzo. Il terzo requisito è un alto throughput, affinché molti computer possano comunicare simultaneamente. La maggior parte delle numerose LAN attualmente sul mercato, pur diverse tra loro, soddisfano ormai questi requisiti.

Poche invece soddisfano il quarto requisito, ossia la capacità di usare dei protocolli (ossia convenzioni) di comunicazione diversi, in modo da poter interconnettere computer tra loro dissimili. Poiché molte organizzazioni considerano l'acquisto di una LAN solo dopo aver accumulato computer di vario tipo, il fatto che la rete non fornisca una adeguata gestione dei protocolli costituisce uno dei maggiori motivi di disappunto.

2. I mezzi trasmissivi

Sono oggi disponibili per le LAN diversi mezzi trasmissivi di basso costo, in diverse gamme di prestazioni. Una possibilità è rappresentata dalla coppia di fili di rame attorcigliati, normalmente impiegata negli impianti telefonici (doppino). Un altro mezzo è il cavo coassiale, diffuso nei paesi con televisione via cavo. Infine ci sono le fibre ottiche, il cui costo è ormai calato a livelli tali da renderle competitive in questo tipo di applicazioni. La scelta del mezzo su cui basare la rete è influenzata da vari fattori di natura più pratica che tecnica. Due fattori sono spesso importanti e cioè la larghezza di banda richiesta dall'applicazione e, poiché alcuni protocolli impiegano una banda più ampia di altri, il protocollo sotto cui opera la rete. Se questi sono i fattori determinanti, bisogna scegliere anzitutto lo schema di modulazione del segnale e poi un mezzo che supporti lo schema di modulazione scelto.

Il più semplice di tutti gli schemi di modulazione è quello in *banda base*. In una rete in banda base il segnale è trasmesso da un nodo direttamente sulla linea, senza essere modulato su una portante. Secondo una nota relazione (di Nyquist), la velocità con cui i dati possono essere trasmessi su un canale di comunicazione è limitata a circa il doppio della frequenza massima del canale. Per esempio, se la frequenza massima è 100 kHz, la velocità massima potenziale è di 200 kbit/sec. In pratica la velocità è influenzata da varii fattori, quale lo schema impiegato per ripartire il tempo di trasmissione.

La modulazione in *banda larga* è uno schema che aumenta il numero di segnali in banda base che un dato mezzo può trasmettere. In una rete a banda larga il segnale in banda base viene modulato su una portante ad alta frequenza prima di essere trasmesso. Poiché il segnale in banda base occupa solo una piccola parte della larghezza di banda della portante, diventa possibile convogliare parecchi segnali in banda base su uno stesso canale. Per esempio, trasmettendo un segnale con banda base di 100 kHz a 100 MHz, esso usa solo l'1% della capacità della portante di trasmettere informazioni. È evidente che tanto maggiore è la frequenza della portante tanto più numerosi sono i segnali in banda base che si possono trasmettere. Mentre la velocità di trasmissione dati in una rete in banda base dipende da quante comunicazioni in banda base il protocollo che controlla l'accesso alla rete riesce a interlacciare, in una rete a larga banda essa è la somma di molti di tali aggregati.

La trasmissione in banda base è impiegata su doppini telefonici e su certi tipi di cavi coassiali e di fibre ottiche. La sensibilità del doppino alle interferenze elettromagnetiche limita sia la frequenza di trasmissione che la distanza del collegamento. Anche se le caratteristiche dei doppini possono differire di molto, una regola generale è che linee costituite da doppini non schermati e senza ripetitori intermedi siano usate per velocità inferiori a un Mbit/sec e distanze non superiori a qualche centinaia di metri. Col cavo coassiale, invece, si possono avere trasmissioni in banda base fino a 10 Mbit/sec su distanze di qualche migliaia di metri. Questa differenza è dovuta alla struttura del cavo, che è costituito da un conduttore centrale separato mediante uno strato isolante dal conduttore cilindrico esterno. Quest'ultimo è messo a terra e agisce da schermo verso le interferenze elettriche, per cui il cavo coassiale risulta meno sensibile ai disturbi del normale doppino.

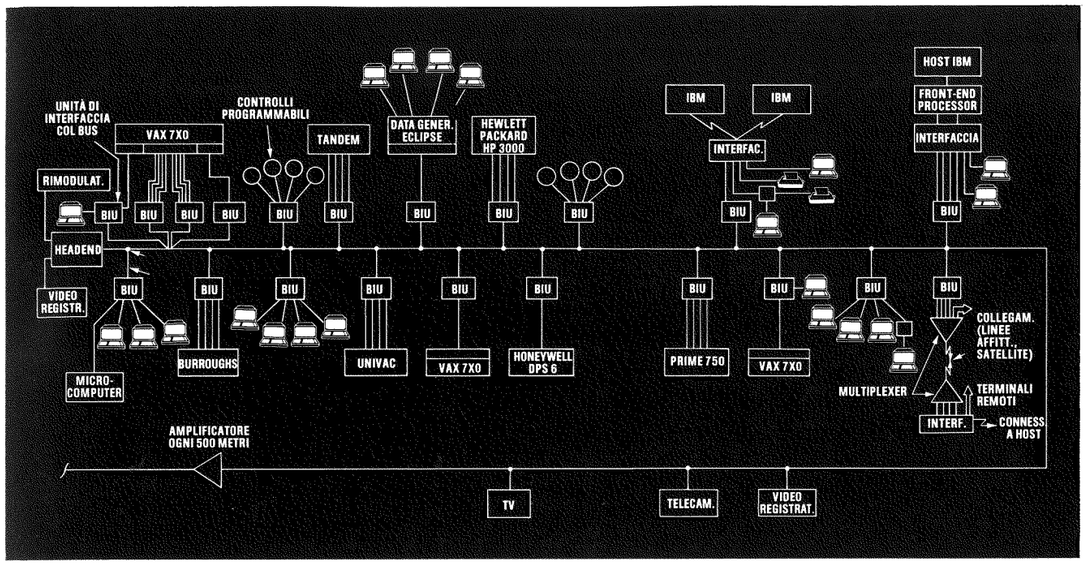


Fig. 1 - Idealmente una rete locale dovrebbe consentire ad elaboratori dissimili di comunicare tra loro e con altre apparecchiature elettroniche di vario tipo. Nello schema qui mostrato, le connessioni alla rete sono realizzate mediante dei dispositivi di interfaccia (BIU = Bus Interface Unit). La figura vuole fornire una indicazione della varietà di compiti a cui una rete locale può essere adibita. In effetti, la maggior parte delle reti attuali offre servizi più limitati.

Il cavo coassiale consente anche la trasmissione in banda larga. Le prestazioni dipendono fortemente dal modo in cui la banda di frequenza viene suddivisa in canali; valori comuni sono velocità da 20 a 30 Mbit/sec su distanze di diversi chilometri. Una rete tipica assegna ai computer uno o due canali da 10 Mbit/sec e inoltre trasmette i segnali di una varietà di dispositivi analogici.

Le fibre ottiche costituiscono un caso a se. Di tutti i mezzi visti, è quello con maggior larghezza di banda potenziale, poiché trasmette nel campo delle frequenze luminose, e risulta inoltre particolarmente immune da disturbi elettromagnetici. Lo sfruttamento di questa capacità trasmissiva implica l'impiego di schemi a banda larga, ma la tecnica del multiplexing ottico è ancora sperimentale. Attualmente, le fibre ottiche sono nate essenzialmente come mezzo per realizzare reti veloci in banda base. Le reti a fibre ottiche esistenti consentono velocità fino a 50 Mbit/sec su distanze di svariati chilometri, ossia cinque volte almeno le prestazioni ottenibili con cavo coassiale in banda base. Tuttavia sono ancora possibili grandi miglioramenti. I più recenti sviluppi di dispositivi ottici trasmittenti e riceventi consentono infatti di inviare dati alla velocità di 12.000 Mbit/sec, su collegamenti in fibra ottica punto a punto.

Reti in fibra ottica con larghezza di banda molto maggiore delle attuali si potranno realizzare con progresso della tecnologia di multiplexing dei segnali ottici. Sono attualmente in sviluppo due soluzioni, dette a divisione di lunghezza d'onda e a divisione di frequenza. Un multiplexer realizzato recentemente dalla Mitsubishi è rappresentativo della prima soluzione. Esso può combinare la luce di due sorgenti che trasmettono su frequenze diverse per inviarle su fibra ottica, e separare la luce che arriva all'altro capo della fibra da due sorgenti addizionali, che trasmettono su frequenze ancora diverse. Occorre tuttavia osservare che tale multiplexer consente al più la trasmissione simultanea sulla

stessa fibra di quattro segnali in banda base, mentre una rete a cavo coassiale può trasmetterne centinaia e anche più.

3. Topologia delle reti

Una volta scelto il mezzo trasmissivo, le rimanenti scelte riguardano la topologia e il protocollo di controllo dell'accesso. Poiché i mezzi trasmissivi sono sostanzialmente a buon mercato e spesso forniscono più larghezza di banda del necessario, le LAN tendono ad usare topologie e strategie di controllo accesso che sprecano una parte della largezza di banda del sistema ma in compenso semplificano l'hardware di comunicazione. Così, ad esempio, la maggior parte delle LAN usa una topologia del tipo ad anello o a bus in cui, poiché tutti i nodi usano lo stesso canale di comunicazione, i messaggi non devono essere fisicamente smistati alle varie destinazioni. Inoltre, sono realizzabili semplici strategie di controllo accesso, in quanto ogni nodo può sondare lo stato del canale comune prima di trasmettere. Una conseguenza è che la connessione alla rete non richiede più di una scheda per nodo. Il costo di connessione risulta quindi sufficientemente basso da rendere ragionevole il collegamento di microcomputer alla rete.

Una rete in cavo coassiale e banda base viene usualmente realizzata con topologia ad anello o a bus. In una rete ad anello di questo tipo, un pacchetto di dati passa da nodo a nodo, con percorso unidirezionale, finché raggiunge il destinatario o finché non ritorna al nodo emittente.

In una rete a bus, invece, i nodi sono collegati da un percorso bidirezionale: i pacchetti, dal punto di inserzione, viaggiano in entrambe le direzioni del bus, fino ai suoi estremi. Delle due topologie menzionate, il bus è quello che consente le più semplici interfacce con le apparecchiature connesse. Nelle reti ad anello, poiché i nodi devono generalmente essere in grado di rimuovere certi messaggi e di mandare avanti gli altri, essi debbono di norma avere la capacità di rigenerare i segnali. Pertanto in un anello ogni nodo include normalmente un ripetitore attivo. In un bus, invece ciò non è richiesto, in quanto i nodi fanno una campionatura del segnale, senza modificarlo.

Quali sono le topologie da usare in corrispondenza dei vari schemi di modulazione e dei vari mezzi? Non

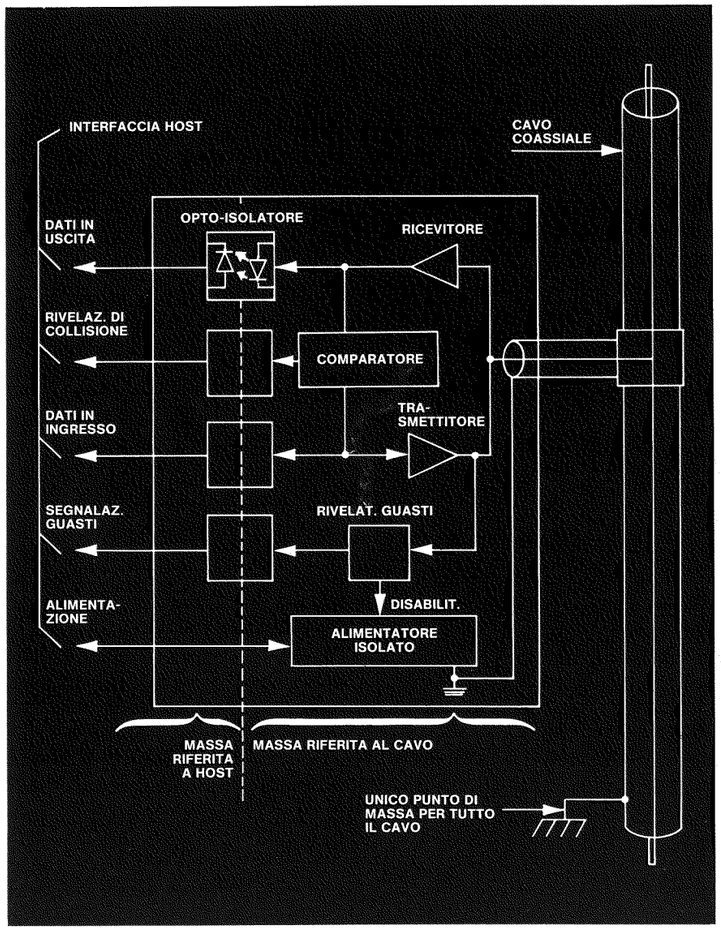


Fig. 2 - Una interfaccia in banda base può essere relativamente semplice. La figura illustra il caso di una rete a bus, con cavo coassiale, operante in banda base sotto un protocollo di accesso del tipo a contesa. (Questo protocollo fa parte degli strati bassi nei modelli di rete, ed è generalmente realizzato in hardware). I circuiti di interfaccia sono divisi in due parti, gli uni connessi alla massa del host, gli altri a quella del cavo coassiale; i dispositivi optoelettronici servono a impedire il passaggio dei disturbi che si possono generare quando si connettono molte apparecchiature ad una massa comune.

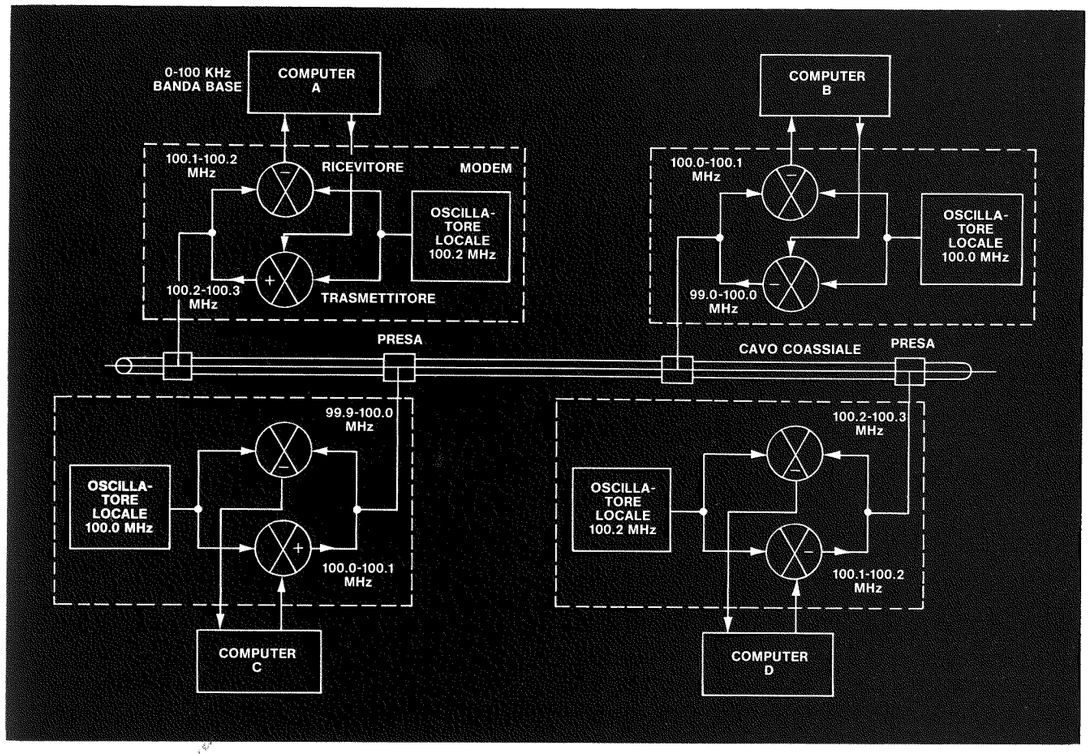


Fig. 3 - Una interfaccia in banda larga include un modem, ossia un dispositivo che modula una frequenza portante con il segnale in banda base, ed effettua l'operazione opposta in ricezione. La modulazione/demodulazione è realizzata sul principio dell'eterodina, ben noto in radiotecnica. Nell'esempio ipotetico qui illustrato vi sono quattro frequenze intermedie, assegnate in modo da consentire la comunicazione bidirezionale tra i computer A e C e tra B e D.

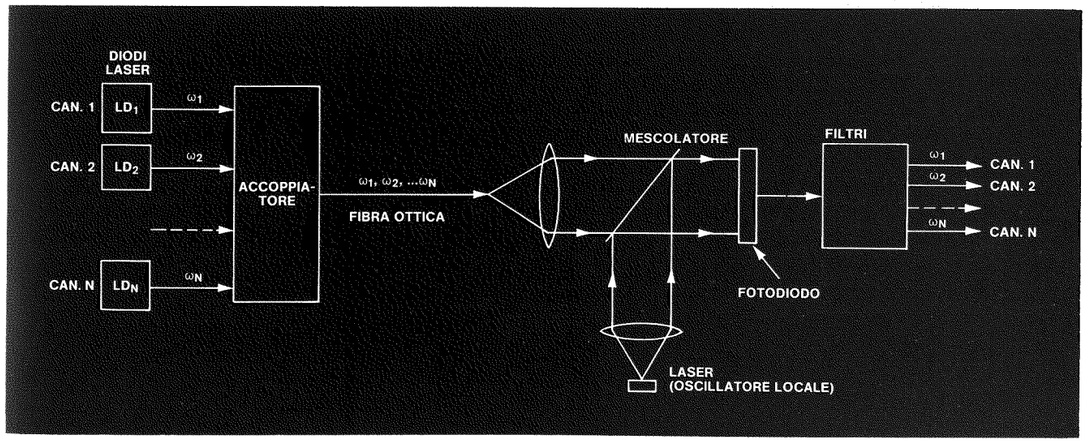


Fig. 4 - La figura mostra lo schema di una rete in banda larga operante con fibra ottica. I laser a diodo sono sintonizzati (termicamente) su frequenze di emissione poco diverse tra loro. Ognuno di essi viene modulato da un segnale in banda base e le frequenze modulate vengono trasmesse simultaneamente sulla fibra ottica. Alla ricezione, la luce viene mescolata con quella di un laser a frequenza fissa, ottenendo una frequenza intermedia diversa per ciascun canale. Queste vengono convertite in frequenze elettriche dal fotodiodo e selezionate successivamente mediante filtri elettronici. Lo schema illustrato è detto a divisione di frequenza. Uno schema alternativo è quello a divisione d'onda, in cui la conversione delle frequenze ottiche in elettriche (ossia il fotodiodo) è spostata più a valle, dopo che i canali sono stati selezionati con filtri ottici.

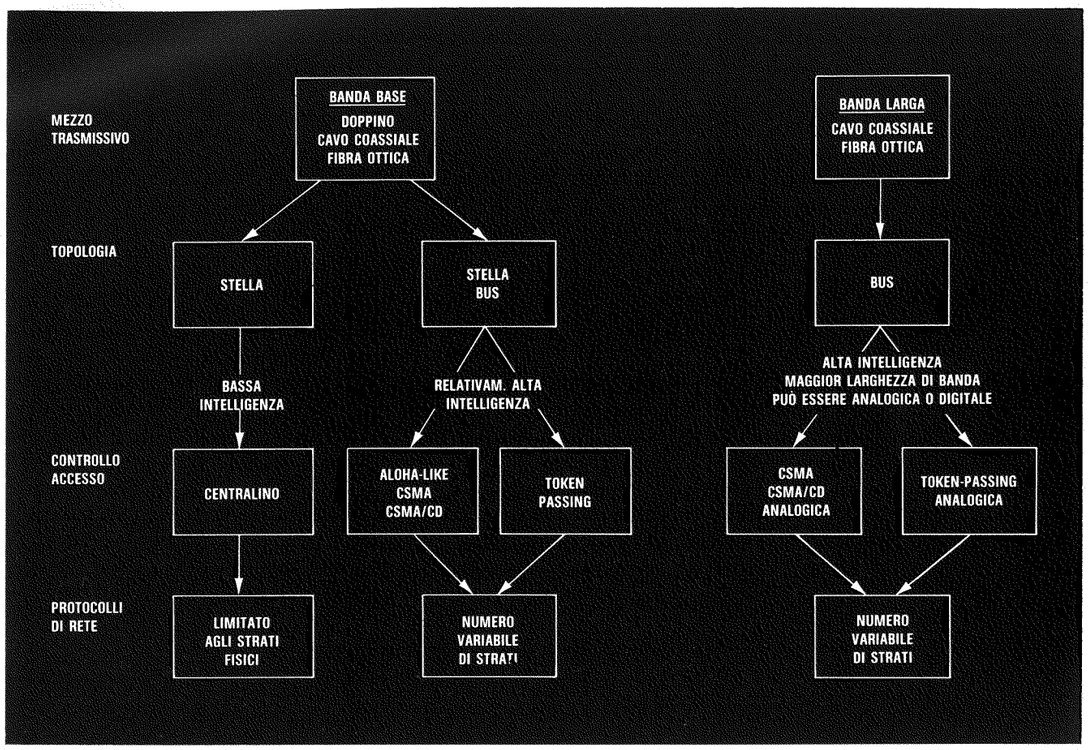


Fig. 5 - Questo schema riassume le più note combinazioni di caratteristiche delle reti locali. In generale la larghezza di banda (ossia la capacità di trasportare informazioni) della rete e la sua intelligenza (o versatilità) aumentano da sinistra verso destra.

vi sono delle regole precise, ma alcune generalizzazioni sono comunque possibili. Molte delle reti in banda base sono realizzate come anelli, anche se non sono per niente limitate a questa topologia. La modulazione a larga banda tende invece verso le topologie a bus, poiché i segnali ad alta frequenza impiegati si propagano uniformemente nel mezzo. Una topologia a larga banda assai usata è il bus con connessioni a stella dipendenti. In una rete di questo tipo, a ciascun ricevitore e trasmettitore viene assegnato un differente canale di frequenze, ed i segnali sono distribuiti ai nodi che costituiscono i vertici di una stella mediante un separatore collocato nel centro di questa. Combinazioni di collegamenti in banda larga e in banda base sono pure fattibili. In un campus universitario, ad esempio, tutte le apparecchiature all'interno di un edificio possono essere connesse tra loro da una rete in banda base e questa, a sua volta, essere collegata a tutte le altre analoghe mediante una connessione in banda larga tra gli edifici.

Reti con fibra ottica in banda base sono raramente realizzate come bus poiché le "prese" ottiche attualmente disponibili introducono notevoli perdite di potenza, limitando severamente il numero di nodi che il bus può servire. Sono invece spesso usate le connessioni a stella in quanto gli accoppiatori ottici più efficienti sono adatti a questa topologia. Le reti a stella presentano, tuttavia, seri inconvenienti, soprattutto una vulnerabilità della rete per quanto riguarda guasti dell'accoppiatore centrale. La topologia migliore per questo tipo di mezzo trasmissivo è probabilmente quella ad anello. Ciò è suggerito anche dal fatto che l'ANSI (American National Standard Institute) sta sviluppando una interfaccia standard per LAN con fibra ottica ad anello. Tale interfaccia dovrebbe definire una rete con velocità di 100 Mbit/sec, la più alta di ogni altro standard attuale.

4. Come ripartire il mezzo trasmissivo

Sia l'anello che il bus presentano il problema di come ripartire fra i vari utenti il mezzo trasmissivo comune. Si tende ad impiegare a questo fine la trasmissione a pacchetto e strategie di controllo accesso coerenti con tale tecnica.

Un modo per introdurre la trasmissione a pacchetto è di confrontarla con la commutazione di circuito, ossia con la tecnica di trasmissione

usata nelle convenzionali reti telefoniche. Durante una normale conversazione telefonica, i periodi di silenzio tendono ad essere corti e poco frequenti; essendo la conversazione tra gli utenti praticamente continua, risulta quindi conveniente stabilire tra essi una connessione fisica, ossia un circuito, dedicando l'intera larghezza di banda alla conversazione per tutta la sua durata. Anche se la commutazione di circuito può essere impiegata per comunicazioni tra computer, questo metodo di trasmissione è assai inefficiente in quanto la comunicazione tra computer tende ad essere intermittente; ad un fiotto di dati segue infatti spesso un lungo silenzio. Perciò, se si assegna un circuito alla connessione, molta della larghezza di banda viene sprecata. Nella maggior parte delle LAN, i dati sono trasmessi perciò a blocchi, o "pacchetti", di dimensioni e formato predefiniti. Sebbene il formato possa differire da rete a rete, come regola generale i dati di un pacchetto sono preceduti da una "testa", che fornisce l'indirizzo del destinatario, e seguiti da una "coda", che contiene dei bit di controllo, che permettono di scoprire gli eventuali errori introdotti durante la trasmissione. Poiché il destinatario di ciascun pacchetto risulta sempre identificato, la rete può essere usata in modo ottimale. Infatti se, per esempio, due computer sono temporaneamente in silenzio, il percorso che li collega può essere utilizzato come transito dei pacchetti provenienti da altre sorgenti e diretti verso altre destinazioni.

Da solo, il concetto di trasmissione a pacchetto è insufficiente a ripartire il tempo di trasmissione tra gli utenti di una LAN. Se gli accessi al mezzo trasmissivo non sono controllati, ci saranno dei momenti in cui più nodi si contenderanno l'uso di tale mezzo, ed altri in cui esso sarà invece inutilizzato. Perciò una rete deve avere una strategia di controllo dell'accesso, che consenta a ciascun nodo di determinare in ogni momento se può trasmettere. Vi sono parecchie strategie a ciò dirette, ma due sono maggiormente usate nelle LAN, e cioé quella della "contention" (contesa) e il "token passing"

Una strategia di controllo del tipo contention è stata impiegata per la prima volta nella rete ALOHA, un sistema di comunicazione a pacchetto via radio realizzato nel 1970 dalla università delle Hawai, per mettere in comunicazione centinaia di terminali distribuiti sulle isole Hawaiane ed un computer centrale collocato a Honolulu nell'isola di Oahu. Anche se la ALOHANET era una rete a larga banda, con tecnica a pacchetti radiotrasmessi, la stessa strategia di controllo accesso venne più tardi impiegata nelle reti locali, incluse quelle in banda base. La rete ALOHANET aveva, come si è detto, un nodo centrale che riceveva i messaggi dai terminali e li ridiffondeva ai medesimi; tuttavia questa stazione funzionava più da ripetitore che da controllore del traffico. I terminali trasmettevano alla stazione di Honolulu su un canale di 100 kHz a 402,35 MHz, mentre la stazione diffondeva le comunicazioni ai terminali su un canale separato, centrato su 413.475 MHz.

La strategia originale di controllo accesso, oggi nota come "puro ALOHA", richiedeva che ciascun nodo quando trasmetteva un pacchetto facesse partire un timer. Alla ricezione di un pacchetto, la stazione centrale eseguiva un calcolo sui bit del medesimo e confrontava il risultato con i bit di controllo errore contenuti nel pacchetto. Se due terminali trasmettevano contemporaneamente, e quindi i due pacchetti interferivano, il confronto precedente falliva, e la stazione centrale rifiutava i pacchetti danneggiati. Se invece la ricezione di un pacchetto aveva successo, la stazione centrale trasmetteva un segnale di riconoscimento a 413,475 MHz; la mancata ricezione di questo segnale prima che il proprio timer finisse di contare, segnalava al terminale trasmittente l'esistenza di una collisione. Il terminale ripeteva allora la trasmissione finché non riceveva il segnale di riconoscimento.

L'ovvio problema con questo schema di controllo accesso è che le collisioni sono probabili in quanto ogni trasmettitore è ignaro degli altri. Un nodo può iniziare a inviare un messaggio quando altri nodi stanno trasmettendo e, anche se c'è collisione, il nodo continua a trasmettere fino alla fine del messaggio. Le ripetizioni di trasmissione, necessarie per recuperare le collisioni, finiscono col consumare molta della larghezza di banda disponibile, riducendo il throughput della rete. In effetti, il puro ALOHA consente un throughput che è circa il 18% della capacità della rete.

Le strategie di controllo accesso denominate CSMA (Carrier Sense, Multiple Access) consentono un throughput più alto. In una rete operante sotto uno dei più semplici algoritmi di questo tipo, un nodo resta in ascolto sulla portante prima di trasmettere. Se il canale è occupato, il nodo aspetta che esso sia libero prima di inserirsi. Perció una collisione può avvenire soltanto se due nodi iniziano a trasmettere circa simultaneamente. Se la collisione accade, il nodo aspetta per un breve tempo prima di riprendere tutto da capo. Il tempo di attesa prima della ritrasmissione ha un valore casuale, in modo da prevenire la possibilità che la stessa collisione accada ancora. Ulteriori miglioramenti si ottengono con gli algoritmi CSMA/CD (Carrier Sense, Multiple Access/ Collision Detection). In reti di questo tipo, la strategia di controllo dei nodi prevede l'ascolto sulla portante sia durante che prima della trasmissione e che il nodo cessi immediatamente di trasmettere se viene rivelata una collisione. In questo modo una minore porzione della larghezza di banda viene sprecata a causa delle collisioni. Questa strategia di controllo è stata ideata nel 1976 dai ricercatori del Centro di Ricerche Xerox di Palo Alto per la rete locale Xerox, denominata Ethernet.

Anche se gli algoritmi di controllo accesso CSMA e CSMA/CD possono consentire dei throughput fino al 85-95% della capacità della rete, il throughput effettivo può risultare assai minore. Ogni fattore che aumenti la probabilità di una collisione riduce il throughput. Per esempio, ciò avviene se aumenta il rapporto tra il tempo di propagazione della rete ed il tempo di trasmissione del pacchetto. Si supponga, infatti, che un nodo stia strasmettendo e

che un altro desideri trasmettere. Quanto maggiore è il ritardo di propagazione dei segnali nella rete, tanto più alta è la probabilità che il secondo nodo trovi che la portante è libera e cominci quindi a trasmettere, causando una collisione. Ragionamenti analoghi si applicano alla lunghezza del pacchetto.

Le prestazioni di una rete del tipo contention possono variare molto anche in funzione del carico. La simulazione mediante computer dei più semplici algoritmi CSMA/CD mostra come, all'aumentare del carico, la capacità della rete spesa per il recupero delle collisioni salga esponenzialmente. Con un carico molto alto, il throughput cade a zero e la rete viene messa fuori uso. Le versioni più sofisticate dell'approccio CSMA/CD tengono conto delle condizioni del carico e forniscono throughput accettabili anche con carichi onerosi. Per esempio, l'algoritmo attualmente impiegato nella rete Ethernet si adatta ad un traffico più pesante aumentando l'intervallo di randomizzazione (ossia il tempo che i nodi devono lasciar passare prima di riprovare a trasmettere) quando il numero di collisioni aumenta. Più che ottenere una disciplina nell'uso della rete mediante un aumento della complessità degli algoritmi è, tuttavia, preferibile segliere una strategia di controllo intrinsecamente più disciplinata.

Una strategia di questo tipo è il token-passing. Un token è semplicemente un insieme distintivo di bit che viene trasmesso da nodo a nodo nella rete. Un nodo della rete con un messaggio da trasmettere aspetta finché arriva il token, esamina i bit che passano e cambia la loro configurazione in un'altra, nota come connessione, che dice agli altri nodi che segue un messaggio. Il nodo trasmette allora il suo pacchetto e ripristina il token. Anche se questo approccio è intrinsecamente semplice, un algoritmo token-passing richiede tuttavia delle strategie piuttosto complicate per il recupero degli errori; infatti se un errore transitorio distrugge un token, può essere difficile determinare che il token è stato perso e decidere quale nodo debba crearne un altro. Una delle prime reti ad impiegare il token-passing è stata la ARC-net della Datapoint, apparsa nel 1977. La Divisione Process Management Systems della Honeywell ha pure sviluppato una rete token-passing, che è ora parte del TDC 3000, il suo più avanzato sistema per il controllo di processo.

Il maggior livello di disciplina delle reti token-passing le rende particolarmente attraenti per la realizzazione di LAN che debbono rispondere a determinati eventi entro un tempo prefissato, come può verificarsi, ad esempio, in una fabbrica. Se il traffico sulla rete è scarso, un messaggio può avere un ritardo maggiore in una rete token-passing che non in una contention. In una rete del primo tipo, tuttavia, il ritardo è fissato dal numero dei nodi, mentre in una rete del secondo tipo esso è variabile e dipende dal traffico. Se un messaggio deve essere inviato entro un tempo predeterminato, ciò può essere di importanza cruciale. In sostanza, il throughput di una rete token-passing è più prevedibile di quello di una rete a contention. Esso dipende essenzialmente dal numero di nodi e dal ritardo introdotto da ciascuno di essi per l'elaborazione del token e per altre routine di controllo, ed è meno sensibile alle caratteristiche proprie della rete, quale il ritardo di propagazione. Esso risente inoltre meno delle condizioni di traffico. Teoricamente, una rete token-passing non può essere messa fuori uso da un carico pesante e le sue prestazioni sono facilmente prevedibili sotto qualunque condizione di carico. Ciò può essere determinante in diversi tipi di applicazione, quali l'automazione di una fabbrica.

Reti del tipo Ethernet sono attualmente ben consolidate, mentre quelle a token-passing sono ancora agli inizi. Una conferma in questo senso è costituita dal fatto che i fabbricanti di dispositivi a semiconduttore producano ormai controlli a basso costo per reti di tipo Ethernet, mentre ciò ancora non accade per le reti token-passing. I meriti di ciascuno di questi due tipi di reti sono comunque tuttora oggetto di discussione. I sostenitori della contention affermano che i problemi elencati per questo tipo di soluzione si applicano solo a certi algoritmi, ma non a tutti. Gli oppositori, invece, confermano che le reti a contention sono imprevedibili se il carico è maggiore del 30% della capacità della rete. Un fattore di confusione è costituito dal fatto che sono attualmente in commercio diversi sistemi tipo ALOHA a basso prezzo, i cui costruttori non forniscono volentieri informazioni sui loro protocolli. Ad ogni modo, qualunque siano i meriti delle due strategie, la tendenza degli utenti è sempre più in favore delle reti a contention per applicazioni di automazione d'ufficio e delle token-passing per l'automazione della fabbrica. Per esempio, la General Motors ha stabilito degli standard per le reti locali acquistate dalle proprie fabbriche (Manufacturers Automation Protocol, o MAP standard), che sono del tipo token-passing, con topologia ad anello.

5. Protocolli di livello più alto.

Consentire che computer tra loro dissimili comunichino facilmente tra loro, è risultato essere un problema più difficile di tutti quelli discussi finora. Si è cominciato a lavorare su questo problema nel 1978, quando l'ISO (International Standards Organization) introdusse il modello OSI (Open System Interconnection) per il collegamento di computer dissimili. Questo modello è stato adottato dallo IEEE (Institute of Electrical and Electronic Engineers), che ne ha sviluppato una versione per reti locali, denominata IEEE-802 standard. (v. figura 6).

Questo modello si propone di rendere più facile il progetto di un sistema complesso di protocolli di comunicazione assegnando le funzioni che la rete deve realizzare a strati separati del sistema. Ciascun strato fornisce dei servizi agli strati superiori, schermando questi ultimi dai dettagli realizzativi di tali servizi. Per esempio, un messaggio generato da un processo di livello superiore di questa struttura può essere compresso dallo strato sottostante e diviso in unità più piccole dello strato successivo. Gli ulteriori strati possono a loro volta attaccare al messaggio la testa o la coda. Lo strato più basso è quello che fisicamente trasmette il messaggio. Evidentemente, il messaggio sarà ricevuto correttamente solo se ciascun strato della apparecchiatura ricevente opera con le stesse regole del corrispondente strato di quella trasmittente, oppure se una interfaccia di rete trasforma un insieme di regole in un'altro. L'obiettivo dell'attività del IEEE è di definire queste regole.

Malgrado l'apparente semplicità del compito, finora sono stati definiti solo gli strati inferiori del modello. Questo evidenzia un problema poiché significa che in effetti i protocolli differiscono da rete a rete e talvolta, per una stessa rete, da un componente hardware ad un altro. Una conseguenza è che i programmi software non possono essere portati da una rete od un'altra o, in una stessa rete, da un computer ad un'altro dissimile.

Un ulteriore problema è che anche quelle reti che sono compatibili con lo standard IEEE offrono solo un sottoinsieme dei servizi che esso definisce. L'elaborazione dei protocolli può essere effettuata da software che gira sugli host computer, oppure dall'interfaccia di rete e dal software associato. Il numero di strati gestiti da una interfaccia di rete ed i protocolli supportati da ciascun strato, pongono delle limitazioni sui tipi di unità che possono essere connesse alla rete e sulle applicazioni cui questa può essere adibita, a meno che l'utente voglia intraprendere una pesante attività di sviluppo di software.

La maggior parte delle LAN sono compatibili soltanto a livello dei primi tre strati del modello IEEE, che possono essere realizzati mediante interfacce hardware. Ciò significa che la maggior parte delle reti locali può collegare soltanto computer di uno stesso produttore. Inoltre, poiché differenti sistemi operativi richiedono differenti protocolli, anche le reti che accettano computer dissimili, possono impiegare certi computer e non altri. Il collegamento alla rete di periferiche o di apparecchiature speciali richiede protocolli software dei livelli superiori, che spesso non sono supportati. Per esempio, collegare dei dischi rigidi o dei controllori programmabili può richiedere dei protocolli dello strato di applicazione che servano come unità di governo e dei protocolli dello strato di presentazione che convertano i dati da un codice ad un altro.

I programmi applicativi pongono ulteriori problemi. Per esempio molte applicazioni richiedono la trasmissione sporadica di brevi blocchi di dati da molti utenti ad uno stesso destinatario piuttosto che lo scambio prolungato di dati tra due utenti. Se la rete deve essere usata per una applicazione di questo tipo, essa deve supportare i protocolli dello strato di trasporto, detti protocolli delle transazioni, oppure l'utente deve scrivere il software di comunicazione da far girare sugli host computer insieme col software applicativo. Applicazioni specifiche, quali lo word processing o il trasferimento di archivi, richiedono pure i servizi degli strati superiori che possono essere non supportati o per i quali differenti fornitori di reti hanno differenti soluzioni.

Anche se non risolve il problema, la definizione degli strati più alti dello standard IEEE sicuramente può giovare. L'attività su questi strati è andata avanti molto lentamente. Un motivo è la strenua resistenza dei produttori alla standardizzazione, essendo loro interesse mantenere un rapporto privilegiato con gli utenti che riescono ad attrarre. Ma non è soltanto il fattore concorrenza a rendere difficile il processo di standardizzazione. I primi tre strati del modello OSI riguardano protocolli hardware che possono essere sistemati mediante interfacce e che richiedono un intervento minimo sui computer. Andando verso i protocolli di livello più alto, può risultare impossibile definire convenzioni così generali che non pongano seri vincoli sulla definizione dei futuri sistemi di elaborazione.

Una ulteriore difficoltà è costituita dal fatto che i protocolli dei livelli superiori possono introdurre dei ritardi sufficientemente ampi da impedire applicazioni in tempo reale. In particolare, vi è la possibilità che essi possano precludere una delle forme più interessanti di sistema distribuito, ossia la base di dati distribuita. Si tratta, infatti, in questo caso di far si che le informazioni memorizzate in molti computer differenti vengano aggiornate simultaneamente da molti utenti. La ricerca, l'aggiornamento e la verifica di informazioni disperse geograficamente, potrebbe richiedere strategie relative al traffico di rete più complesse di quanto il modello IEEE sembra consentire. Per ragioni analoghe, gli standard di livello superiore potrebbero precludere la comunicazione vocale digitalizzata o la trasmissione di segnali televisivi compressi per videoconferenze. Per finire, nessuno sa ancora bene dove la posta elettronica si inserisca nella struttura a sette strati. In effetti, il National Bureau of Standards ha proposto un differente gruppo di protocolli per risolvere questa applicazione. Si sta inoltre lavorando su uno standard per posta elettronica avanzato dal CCITT, l'organizzazione internazionale responsabile degli standard di telecomunicazione.

Malgrado le difficoltà, la definizione di uno standard è di importanza cruciale, non solo perché esso renderebbe più facile la connessione alla rete locale di apparecchiature dissimili, ma anche perché faciliterebbe la connessione di reti tra loro diverse. Se i protocolli fossero compatibili, le reti potrebbero essere collegate con dei semplici ponti, essenzialmente dei ripetitori intelligenti dotati di una "funzione di filtro" che determina quale dei pacchetti ricevuti da una rete deve essere inviato ad un'altra. In una grande organizzazione, ogni dipartimento potrebbe così scegliere la tecnologia di rete più adatta alle proprie necessità e tuttavia condividere le risorse con altri che hanno fatto scelte differenti. Inoltre, ciò consentirebbe di gestire economicamente la crescita del traffico di rete. Infatti, se il traffico aumenta e le prestazioni di una rete locale tendono a deteriorarsi, il carico potrebbe essere suddiviso tra due reti tra loro interconnesse.

6. Alternative alle reti locali.

In questo articolo si è parlato ci ciò che comunemente viene chiamata una rete locale di elaborazione, ma

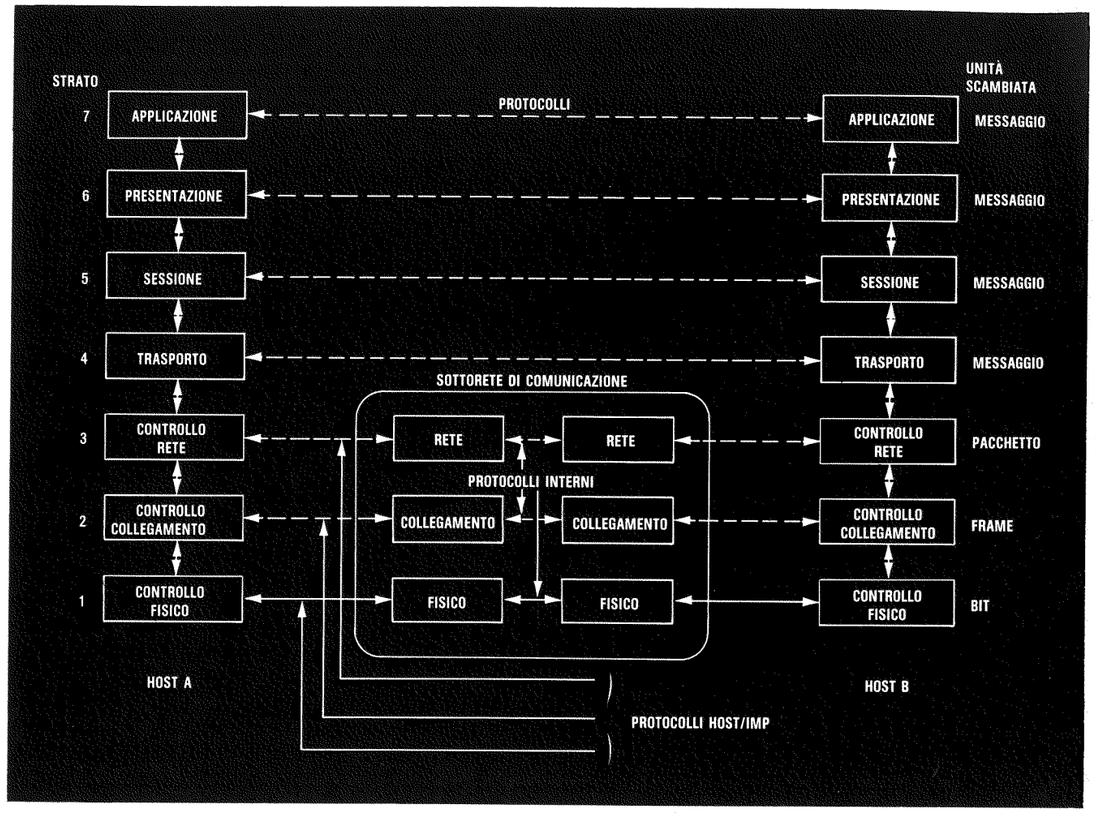


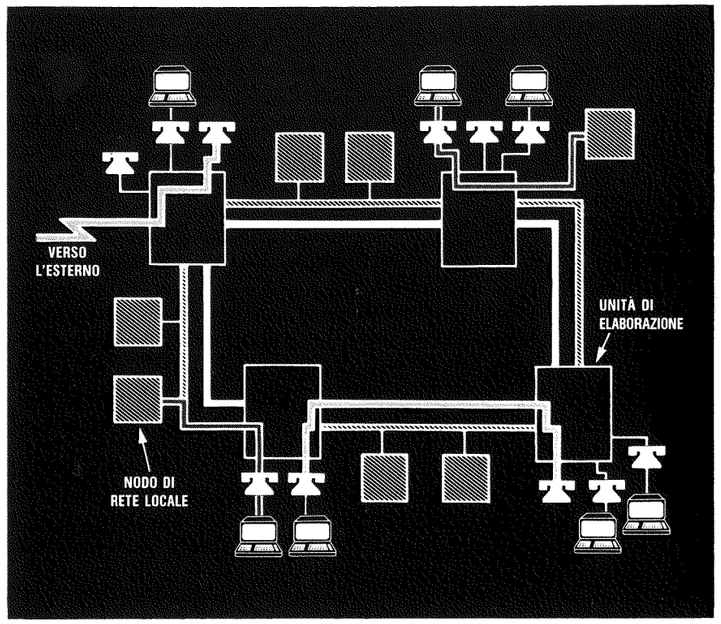
Fig. 6 - Lo standard IEEE-802, qui schematizzato divide concettualmente una rete locale in strati, ciascuno dei quali trasforma i dati in un modo prescritto. Un messaggio viene ricevuto correttamente solo se le trasformazioni che avvengono al nodo ricevente sono esattamente, ma nell'ordine opposto, quelle del nodo emittente. L'attività di standardizzazione consiste nel definire le convenzioni, o protocolli, per ciascun strato. I primi tre strati contengono i protocolli di interfaccia con la sottorete di comunicazione e vengono generalmente realizzati in hardware (IMP: Interface Message Processor). Gli strati superiori sono invece di norma realizzati in software, di dotazione degli host computer. Poiché molte reti non sono conformi allo standard e per il fatto che i protocolli degli strati superiori non sono stati ancora definiti completamente, ciò che funziona su una rete generalmente non funziona su un'altra.

occorre osservare che esistono soluzioni alternative per realizzare la comunicazione dati su brevi distanze. Due esempi di alternative alle reti locali sono costituite dai PBX e dalle reti a pacchetto via radio.

Il PBX (Private Branch Exchange) per dati rappresenta la controproposta della comunità telefonica alle reti locali. Come il PBX per la voce, si tratta essenzialmente di un centralino che stabilisce le connessioni tra i rami che in esso confluiscono. Essendo la frequenza dei segnali più elevata, i PBX per dati consentono velocità di comunicazione molto maggiori dei PBX vocali. Anche se il PBX per dati è relativamente lento e non supporta protocolli superiori al terzo strato del modello IEEE, esso offre tuttavia una buona larghezza di banda a costi minori delle altre reti locali. La prossima generazione di PBX, cioè i PBX per voce e dati ed i PBX a banda larga, potrà probabilmente competere con successo, se non altro perché molti utenti richiedono comunicazioni sia vocali che di dati, ed una rete è preferibile a due. In sostanza, è improbabile che una sola tecnologia possa dominare completamente il mercato. Il futuro può forse essere meglio individuato in soluzioni ibride quali stanno apparendo, cioè PBX per voce e dati che possono comunicare tra loro attraverso reti locali e, parallelamente, reti locali che possono trattare i PBX come nodi.

Un'altra soluzione interessante è costituita dalle reti a pacchetto via radio. La divisione della Honeywell che si occupa di prodotti e servizi per gli edifici sta lavorando ad una rete pilota che dovrebbe consentire al personale di manutenzione che è fuori sede di accedere ad una base di dati su parti, clienti e contratti, mentre dovrebbe permettere al personale di vendita di valutare i risparmi energetici ottenibili con determinati

Fig. 7 - Questo è un esempio di soluzione "ibrida", che combina cioè reti locali con centralini telefonici (PBX). In questa realizzazione della Ztel Inc. di Wilmington (Massachusetts), vi è una rete ad anello (tratteggiato in figura), a pacchetto del tipo token-passing, operante ad alta velocità, ed una rete pure ad anello (non tratteggiato), a commutazione di circuito, su cui sono trasmessi a bassa velocità dati e voce (con sistema PCM). Come mostrano le connessioni della parte inferiore della figura, una richiesta di dati da un microcomputer ad un computer collegato alla rete locale viene inviata attraverso l'anello token-passing, mentre una chiamata telefonica da un posto ad un altro viene effettuata attraverso l'anello a commutazione di circuito. Lo schema minimizza il numero di fili necessari per connettere un sistema con molti nodi.



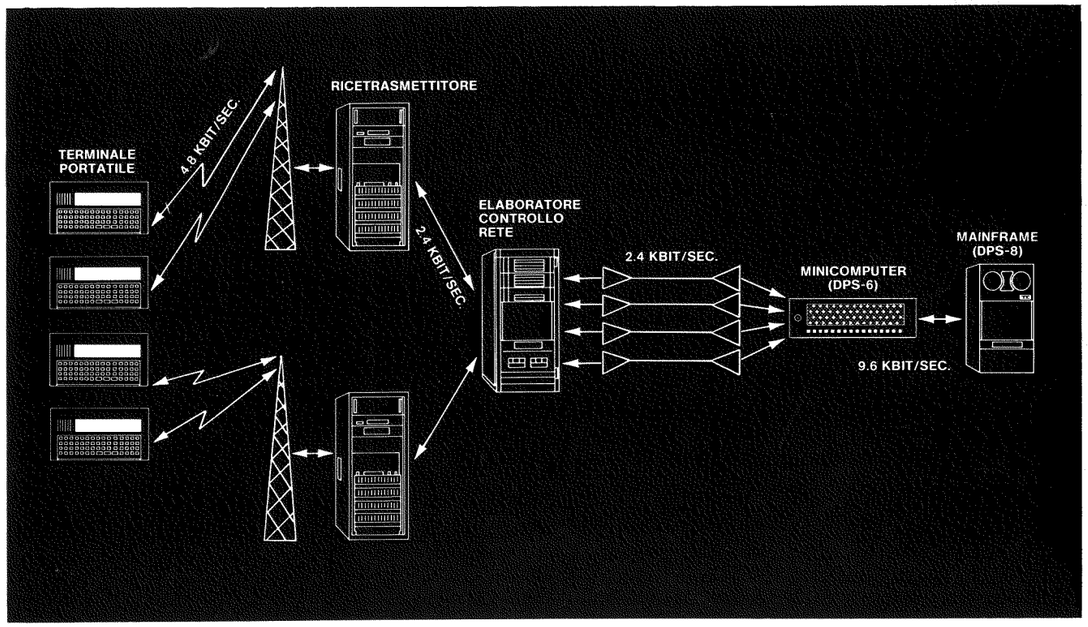


Fig. 8 - Schema di principio della rete Cricket, una rete sperimentale della Honeywell, progettata per fornire ad utenti mobili l'accesso ad una base di dati con informazioni commerciali, residenti su un mainframe (DPS-8). I computer portatili, con ricetrasmettitore incorporato, comunicano mediante trasmissione a pacchetto, sulla banda commerciale di 800 MHz, con diverse antenne. I segnali vengono inviati, attraverso dei controllers, ad un minicomputer (DPS-6), che funziona da front-end del mainframe, il quale invia le risposte. Gli eventuali errori introdotti dalla trasmissione via radio sono verificati dai ricetrasmettitori delle antenne (ciò riduce circa a metà la velocità effettiva di trasmissione dei dati). Il minicomputer, oltre che come front-end, può agire da interfaccia verso una rete a lunga distanza; ciò consentirà la interconnessione di reti locali, come quella mostrata, sull'intero territorio.

impianti e presentare proposte agli utenti. I nodi della rete sono costituiti da computer portatili, con un ricetrasmettitore incorporato. Essi sono collegati, con trasmissione a pacchetto sulla banda commerciale di 800 MHz, con alcune antenne radio che coprono il territorio assegnato. Queste, a loro volta, comunicano mediante un controller con un mainframe Honeywell, un DPS 8. Questa rete pilota, denominata Cricket, è limitata per il momento ad una zona; se i risultati saranno soddisfacenti, si realizzeranno reti simili in differenti aree, collegate tra loro da una rete dati su lunga distanza. In sostanza, l'obiettivo è una base di dati distribuita, con accesso e manutenzione attraverso reti locali tra loro connesse.

7. Conclusioni.

In questo articolo si è fatto il punto sulle reti locali di elaboratori, con un cenno anche a possibili alternative delle soluzioni oggi consolidate. Le reti locali attuali realizzano efficacemente alcuni obiettivi, ma per sfruttare completamente la potenzialità dei piccoli computer, sarà necessario impiegare diversi altri modi e strumenti di comunicazione.

Correzione automatica di errori ortografici

GABRIELLA CAMPI

Honeywell Information Systems Italia
Roma

1. Introduzione

L'automazione di un ufficio nasce in particolare dall'esigenza sempre più pressante di snellire una gran mole di lavoro in tempi sempre più brevi.

Questo tipo di esigenza crea notevoli problemi, tra cui l'interazione di utenti inesperti con apparecchiature sempre più sofisticate. Nasce pertanto la necessità di studiare interfacce molto evolute, tali da permettere l'utilizzo dei mezzi di elaborazione dei dati da parte di utenti che non hanno una cultura specifica in merito. Il termine "automazione d'ufficio" indica un processo che, partendo dall'ufficio classico, tende a farlo evolvere verso un tipo di ufficio in cui ogni attività viene svolta col supporto dell'elaboratore.

Dall'esame delle attività svolte dal personale in un ufficio e dei costi relativi, si possono individuare tre aree fondamentali su cui intervenire con l'automazione:

a) Comunicazioni: una esigenza molto viva è rappresentata dal miglioramento dei processi di trasmissione e ricezione di messaggi e documenti, sia scritti, comprendenti anche grafici, sia orali; possiamo individuare in questo ambito quattro filoni fondamentali:

- posta elettronica: si intende la trasmissione di messaggi interamente composti di testo scritto;
- teleconferenza: sostituisce le riunioni classiche permettendo l'interazione a distanza di un gruppo di persone;
- trasmissione di facsimile: una pagina, eventualmente comprendente anche delle figure, viene digitalizzata e poi trasmessa;
- interazione col telefono.

b) Trattamento dei documenti: nasce dall'esigenza di possedere apparecchiature atte a consentire una rapida stesura, correzione e stampa multipla dei documenti stessi.

c) Archiviazione dell'informazione: un sistema elettronico consente una ricerca rapida e completa delle informazioni cercate.

In questa sede ci occuperemo dettagliatamente di alcuni aspetti riguardanti il punto b), facendo una panoramica generale sullo stato dell'arte e descrivendo i risultati scaturiti da una ricerca personale per la correzione di errori ortografici relativi alla lingua italiana.

Per la preparazione dei documenti vengono utilizzati ampiamente gli elaboratori elettronici, con appositi programmi che facilitano la redazione e la formattazione dei testi ("text editing/formatting").

L'utente può mandare il documento direttamente in linea con il "text editor" e memorizzarlo in un file assieme ai comandi di formattazione, che specificano in modo appropriato i margini, le spaziature, il tipo di impaginazione, l'identatura ecc. Il "text formatter" interpreta questo file per produrre un file di uscita pronto per la stampa.

Generalmente il text editor è un programma interattivo, mentre il formatter molto spesso è un programma "batch".

Originariamente questi mezzi per la preparazione di documenti erano usati solo all'interno di ambienti informatici, ma l'analisi dei vantaggi inerenti ad un tale modo di operare ha creato successivamente un mercato per i sistemi di "word-processing".
Con la preparazione automatica dei documenti diventano possibili altre operazioni sul testo, oltre a quelle di editing e di formatting. In particolare, un'analisi per scoprire eventuali errori di ortografia ("spelling"); quest'ultima è una forma di "text-processing" abbastanza recente.

Bisogna dire per prima cosa che esistono due tipi di programmi per l'ortografia: programmi di verifica ("spelling-checker") e programmi correttori ("spelling-corrector").

Il problema per un spelling-checker è abbastanza semplice: dato un determinato testo, esso deve individuare le parole che non sono scritte correttamente.

La funzione di uno spelling corrector invece è più complessa; non solo si devono individuare le parole scritte in modo errato, ma si deve cercare anche di determinare la parola scritta in maniera corretta che con maggiore probabilità si avvicina alla parola che si voleva intendere.

Dato un certo alfabeto, una parola può essere intesa come un punto in uno spazio multi-dimensionale costituito da sequenze di lettere. Non tutte queste sequenze costituiscono

parole di senso compiuto; per cui lo spelling checker deve classificare le sequenze secondo che esse corrispondano a parole corrette o meno. Nella seconda fase, quella più propriamente di spelling correction, si devono collezionare, nello spazio delle parole, tutte quelle più vicine alla parola data, per scegliere quella più probabile.

2. Correzione ortografica dell'inglese

Il primo lavoro sulla correzione ortografica risale al 1957.

In quell'anno Glantz [1] scrisse un articolo in cui metteva a fuoco il problema del riconoscimento delle parole errate in un testo inglese; usando un dizionario di parole corrette e confrontando due stringhe tra loro, cercava di trovare il maggior numero di caratteri uguali in posizioni uguali.

Questo approccio, però, si rivelò utile solo per i lettori ottici di caratteri (OCR), perchè gli errori introdotti da tali periferiche sono soltanto errori di sostituzione di una lettera con un'altra.

Successivamente, nel 1960, Blair [2] propose un metodo in cui le lettere di una parola venivano pesate in modo diverso, a seconda della posizione occupata all'interno della stringa e a seconda del loro tipo, cioè vocale o consonante, per creare delle abbreviazioni di quattro o cinque lettere per ogni parola. Se le abbreviazioni corrispondevano, due parole venivano assunte come uguali.

Nel 1964 c'è finalmente una svolta fondamentale nello studio della correzione di errori ortografici.

Damerau [3] pubblica i risultati di una indagine statistica da cui risulta che l'80% degli errori introdotti in un testo sono generati da parole corrette applicando una delle seguenti regole (fig. 1):

a) trasposizione di due lettere

b) aggiunta di una lettera;

c) omissione di una lettera;

d) introduzione di una lettera errata

Inizia così un approccio più sistematico al problema, che lo svincola da un contesto specifico e permette la realizzazione del primo "spelling checker": lo SPELL, realizzato da Gorin a Stanford nel 1971.

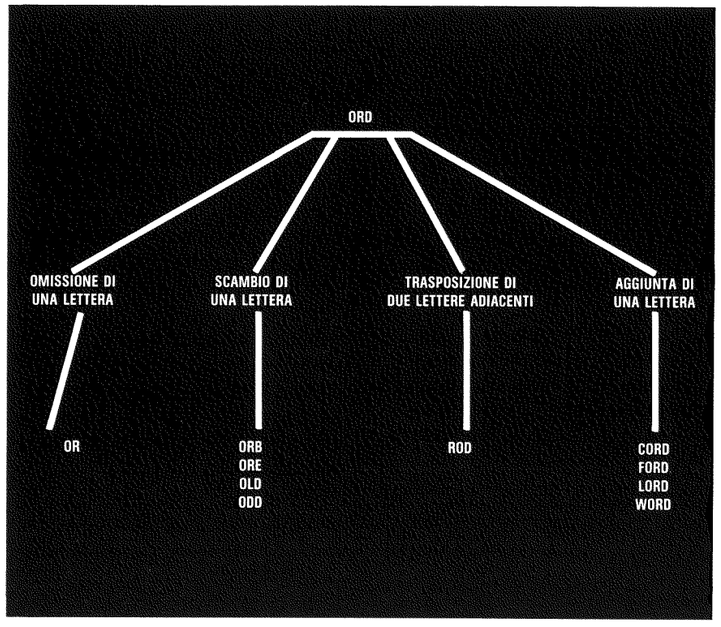


Fig. 1 - I quattro principali tipi di errori ortografici.

L'algoritmo principale dello SPELL per la correzione, è il seguente.

Per ogni "token" (parola potenzialmente errata) che non viene trovato nel dizionario di base, viene costruita una lista di tutte le parole che potrebbero produrlo con l'applicazione di una delle quattro regole precedentemente menzionate. Questa è la lista delle parole candidate alla correzione.

Se la lista contiene una sola parola, essa costituisce la correzione proposta e viene chiesto all'utente se ciò è esatto.

Se la lista contiene più di una parola, l'utente può richiedere la lista e selezionare una delle seguenti opzioni:

1) *sostituzione:* il token sconosciuto è ritenuto errato e deve essere sostituito; esso viene cancellato e la parola corretta viene richiesta all'utente;

2) *sostituzione e memorizzazione:* il token sconosciuto viene sostituito da una parola specificata dall'utente e tutte le volte che questo token sarà usato successivamente verrà sostituito dalla nuova parola;

3) *accettazione:* il token viene considerato corretto e non viene eseguita nessuna operazione specifica;

4) *accettazione e memorizzazione:* come al punto 3); con l'aggiunta che ogni volta che il token verrà trovato successivamente sarà considerato esatto;

5) *edit:* si entra in un ambiente di editing in modo da permettere che il file venga modificato nel contesto ove si trova il token errato.

La lista delle parole candidate alla correzione viene realizzata mediante ricerche multiple nel dizionario.

Ad esempio, la trasposizione di due lettere adiacenti, può essere corretta generando tutte le possibili trasposizioni di due lettere e se la parola viene trovata tra uno dei token risultanti, essa è una candidata da aggiungere alla lista; per un token di n caratteri sono necessarie n-1 ricerche nel dizionario.

La correzione di una carattere aggiunto richiede la cancellazione di un carattere alla volta nel token, e successivamente la ricerca nel dizionario degli n token prodotti; tale algoritmo dà luogo a n ricerche.

La correzione di una lettera errata prevede la sostituzione di ogni carattere potenziale (cioè appartenente all'alfabeto), in ogni posizione all'interno della stringa. Così con N caratteri potenziali, occorrono $N \cdot n$ ricerche nel dizionario.

Per quanto riguarda l'ultimo tipo di errore, cioè l'omissione di un carattere, vengono prodotte $n + 1$ copie del token, inserendo ogni volta uno spazio (blank) in una posizione sospetta; poichè il blank è un carattere che non è contenuto in nessuna stringa, i token così formati vengono trattati con la routine usata per le parole con un carattere errato. Questo procedimento non termina alla prima parola trovata, ma dopo aver cercato il maggior numero possibile di parole potenzialmente corrette.

Tale approccio richiede $n + 1$ ricerche per l'omissione di una lettera e n per una lettera errata.

La velocità di ricerca dipende non soltanto dal tipo di strategia adottata per la ricerca stessa, ma anche dalla struttura e dalle dimensioni del dizionario.

Il dizionario utilizzato dallo SPELL, costruito in base studi statistici, effettuati per determinare la frequenza dei termini costituenti il lessico inglese e riportati nel Brown Corpus (Kucera e Francis 1967), è strutturato secondo tavole per una ricerca rapida ("hash") con 6760 entrate.

La funzione di ricerca per un token è calcolata sulle prime due lettere e sulla lunghezza n del token. Il valore di questa funzione rappresenta il puntatore ad una lista concatenata di parole che hanno la stessa lunghezza e iniziano con le stesse prime due lettere.

Potendo mantenere il dizionario in memoria centrale, la velocità di ricerca viene incrementata notevolmente.

A questo fine è stata messa a punto una strategia per diminuire lo spazio di memoria occupato dal dizionario.

Una prima riduzione si può fare tenendo conto che il 95% del comune lessico inglese viene coperto con circa 16.000 parole, mentre il 90% con solo 8.000 parole.

Una ulteriore drastica riduzione di spazio può essere ottenuta scomponendo le parole in radice e affissi (prefissi e suffissi), mediante regole abbastanza complicate che permettono sia la scomposizione che la ricomposizione della parola, dando così la possibilità di memorizzare solo le radici (fig. 2).

Questa tecnica per la memorizzazione di un dizionario è usata anche dallo "spelling checker" fornito dal sistema operativo UNIX, il quale però adotta tecniche diverse dallo SPELL per l'evidenziazione di una parola errata in un testo.

Il programma TYPO del sistema operativo UNIX, si basa sui risultati ottenuti dalla ricerca sulla frequenza di digrammi e trigrammi (gruppi di due o tre lettere) in un testo inglese.

Con un alfabeto di 28 lettere (alfabetiche, spazio e apostrofo), ci saranno 28^2 (784) digrammi e 28^3 (21.952) trigrammi. La loro frequenza varia notevolmente; tipicamente in un testo soltanto 550 digrammi (70%) e 500 trigrammi (25%) sono effettivamente ricorrenti.

Se un token contiene molti digrammi o trigrammi rari, è potenzialmente errato.

Questa tecnica è una delle più frequenti nella letteratura sulla rilevazione e correzione di errori ortografici (v. bibliografia da [4] a [12]).

Il programma TYPO calcola la frequenza di digrammi e trigrammi in un testo e costruisce una lista dei token contenuti nel testo stesso. Poi, per ogni token calcola un indice di peculiarità. Questo indice rappresenta una misura statistica della probabilità che il trigramma xyz sia stato prodotto dalla stessa sorgente che ha prodotto il resto del testo. Come risultato si ottiene una lista di token ordinati in base all'indice di peculiarità decrescente; siccome le parole errate tendono ad avere un alto indice di peculiarità, esse appaiono all'inizio della lista. Questo numero di token improbabili viene ulteriormente ridotto mediante il confronto di ogni token con una lista di circa 2500 parole comuni; se qualche token viene trovato in questa lista è riconosciuto come esatto ed è eliminato dalla prima lista. I token rimanenti vengono poi rimandati all'esame da parte dell'autore del testo.

Il problema della correzione di errori ortografici non è stato trattato solo da un punto di vista applicativo, con la realizzazione di "spelling checkers", ma anche da un punto di vista puramente teorico, dando luogo a risultati che mettono in rilievo alcune caratteristiche intrinseche della struttura di una stringa e delle tecniche di correzione con confronti stringa a stringa.

Nel 1974 Wagner e Fisher [15] scrivono un articolo, in cui mettono in luce il fatto che il problema della correzione stringa a stringa si può ricondurre alla determinazione della "distanza" tra due stringhe. Essa viene misurata dalla sequenza di operazioni di edit necessarie per cambiare una stringa in un'altra.

In base a tre delle operazioni di edit che Morgan [16] definisce applicabi-

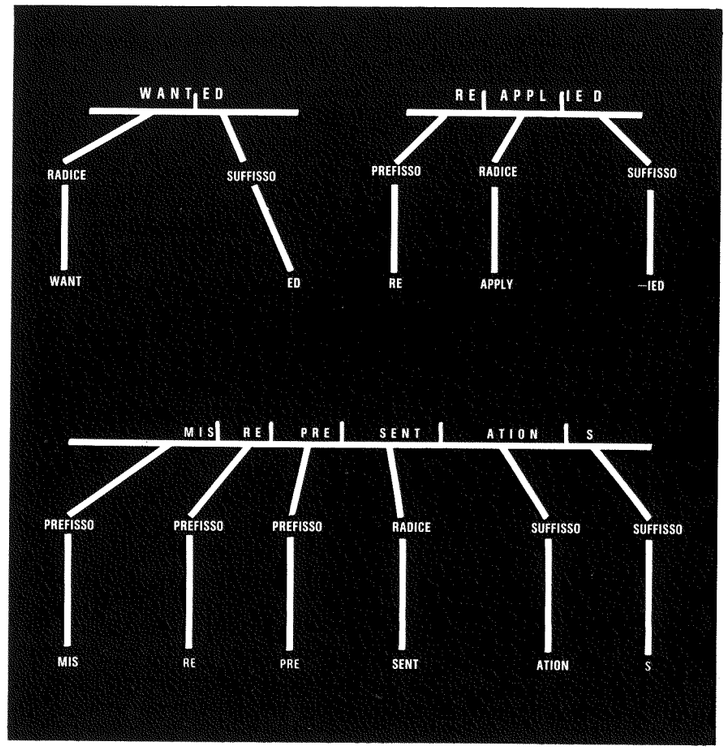


Fig. 2 - Scomposizione di una parola in radice e affissi.

li ad una stringa (scambio di un carattere in un altro, cancellazione di un carattere e inserzione di un carattere), essi definiscono una nozione generale di "distanza" tra due stringhe. Presentano inoltre un algoritmo per computare la distanza, che lavora in tempo proporzionale alle lunghezze delle due stringhe. Questo algoritmo può essere usato, come caso speciale, anche per trovare la più lunga sottosequenza comune alle due stringhe date.

Anche un indice associato ai vari caratteri potrebbe essere utile per la "spelling correction", se si pensa ad esempio che, a causa della particolare disposizione dei caratteri sulla tastiera, il carattere A viene molto più facilmente scambiato con "S" anzichè con "Y".

L'anno successivo, cioè nel 1975, Wagner cerca, insieme a Lowrance, [17] di migliorare i risultati ottenuti.

Essi scrivono un articolo in cui viene ampliato l'insieme delle operazioni di edit eseguibili su una stringa. Introducono infatti l'operazione di trasposizione di due caratteri. Mostrano inoltre che, sotto certe restrizioni dei costi delle operazioni di edit, il problema esteso può ancora essere risolto in un tempo proporzionale alla lunghezza delle due stringhe.

Partendo da questi risultati, nel 1976, Wong e Chandra [18] dimostrano che se le operazioni sui simboli di due stringhe sono ridotte soltanto a test di uguaglianza, allora un numero 0(mn) di operazioni è necessario e sufficiente per computare la distanza tra due stringhe. Cioè, viene stabilito un estremo inferiore, la cui validità è però limitata al tipo di algoritmo definito dagli autori.

Contemporaneamente Teruo Okudo ed altri [19] propongono un nuovo metodo per la correzione di errori ortografici, basato sulla "distanza pesata" di Levenshtein. È questo un modo di calcolare la distanza tra due parole, di cui una derivata dall'altra mediante un certo numero di sostituzioni, inversioni e cancellazioni di lettere, tenendo conto del peso di ciascuna di queste tre operazioni.

Gli autori propongono anche una realizzazione hardware del metodo di correzione basato sulla distanza pesata, ma esso risulta abbastanza complicato e poco flessibile.

Per concludere questa sintesi sullo stato dell'arte, bisogna citare il lavoro presentato da Miller [20] nel 1980, EPISTLE, un sistema per l'analisi automatica della corrispondenza di ufficio, nel quale si passa dal livello di correzione puramente ortografica a livelli superiori, cioè grammaticali e sintattici.

Per quanto concerne la sintassi, EPISTLE riesce ad assegnare strutture corrette a circa il 70% di tutte le costruzioni grammaticali inglesi e a circa l'85% di quelle relative a testi commerciali.

Rispetto alla grammatica, riesce a identificare 14 classi di errori grammaticali (es. discordanza numerale tra soggetto e verbo) e suggerisce le correzioni più probabili. Vengono analizzate anche alcune forme stilistiche e infine l'approccio alla semantica è realizzato mediante l'assegnazione di qualche dozzina di caratteristiche "semantiche", o "primitive", quali ad esempio caratteristiche lessicali.

EPISTLE comunque, non rappresenta un punto di arrivo ma un trampolino di lancio verso orizzonti sempre più vasti, che sconfinano nell'analisi della "semantica intenzionale", un campo, oggi, ancora tutto da esplorare.

3. Correzione ortografica dell'italiano: una soluzione.

Come visto precedentemente, il problema della correzione di errori ortografici è stato affrontato essenzialmente in relazione alla lingua inglese.

L'italiano come l'inglese è una lingua flessiva, cioè la radice di ogni vocabolo ne contiene il significato sostanziale e subisce modificazioni per indicare le relazioni diverse, oppure riceve delle parti modificabili, che per se stesse non hanno alcun valore, mentre, congiunte con la radice, formano un tutto unico. Rispetto alla lingua inglese, l'italiano ha però dei fonemi diversi. Questo significa, ad esempio, che nella lingua italiana non troviamo quasi mai errori di assonanza fonetica, come spesso avviene, invece, nella lingua inglese.

In questa sede proporremo degli algoritmi per la correzione di errori ortografici, applicati alla lingua italiana. Essi non fanno riferimento ad algoritmi presenti in letteratura, perchè si è voluto affrontare il problema con un approccio nuovo, più generale, che tenga conto delle caratteristiche strutturali delle parole, piuttosto che della parola come appartenente ad una determinata lingua.

Gli errori presi in considerazione sono i seguenti:

1) trasposizione di due lettere adiacenti;
2) omissione di una lettera;
3) raddoppiamento di una lettera
4) scambio di una lettera con quella di un tasto adiacente;
5) introduzione di un carattere extra, adiacente al tasto del carattere battuto.

Gli algoritmi sono stati verificati su di un vocabolario di 11.695 termini italiani di uso più comune; la loro capacità di risoluzione è stata del 100%, in tempi dell'ordine dei millesecondi.

3.1. Struttura del dizionario

Gli algoritmi che descriveremo utilizzano, come si è detto, un dizionario di 11.695 termini italiani, di uso più comune, estratti da giornali diversi, in modo da avere termini di uso corrente relativi a molti settori della vita quotidiana.

Si sarebbe potuto costruire il dizionario anche utilizzando il "Lessico di frequenza della lingua italiana" [21], ma il primo era già in versione leggibile dal calcolatore, perciò è stato più comodo utilizzare tale versione.

I termini sono stati ordinati in base alla lunghezza, che va da un minimo di quattro caratteri ad un massimo di 10.

Ogni gruppo di termini, di una data lunghezza, è stato ulteriormente messo in ordine alfabetico e ordinato in base alla frequenza delle lettere distinte che compongono le parole.

Il dizionario viene caricato in memoria centrale in modo da abbassare i tempi di ricerca, i quali vengono ulteriormente ridotti con una tecnica di accesso a due livelli. È stata creata a questo scopo una matrice bidimensionale (LUPLED), in base a due parametri: la lunghezza della parola e il numero di lettere distinte all'interno della parola stessa (fig. 3).

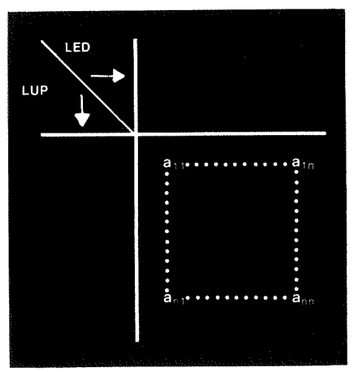


Fig. 3 - Dizionario dei termini registrato in memoria.
LUP = lunghezza della parola;
LED = numero di lettere distinte della parola.

I termini a_{ij}, scelti secondo particolari criteri che verranno descritti in seguito, forniscono l'intervallo entro cui effettuare la ricerca. Questa è di tipo sequenziale e, supposto che abbia successo e che le frequenze di accesso ai nomi siano uniformi, il tempo medio di effettuazione risulta uguale a: (n+1)/2, dove n rappresenta l'estremo superiore dell'intervallo entro cui si effettuano le ricerche.

3.2. Analisi della parola

Data una parola, questa viene analizzata da una apposita routine in modo da costruire un vettore caratteristico contenente dei valori che identificano univocamente la parola.

La routine utilizza, a questo scopo, un vettore ALFAB contenente: 26 caratteri alfabetici, 5 caratteri speciali e le cifre da 0 a 9; un vettore contatore CONT e un vettore posizione POS.

Ogni carattere della parola data, viene confrontato con ALFAB. Se un carattere non viene trovato, viene data una segnalazione d'errore; altrimenti viene incrementato il contatore CONT di 1, nella posizione corrispondente a quella che occupa il carattere trovato nel vettore ALFA B. Dopo di che viene verificato il valore CONT; se è uguale ad 1 allora POS viene posto uguale ad 1. Cioè in POS mettiamo il valore che esprime la posizione del carattere nel vettore ALFAB.

Questo procedimento viene ripetuto fino a che non sono stati scanditi tutti i caratteri che compongono la parola.

I valori contenuti in POS e in CONT vengono utilizzati per creare i seguenti vettori:

1) IND: contiene la posizione alfabetica (cioè in ALFAB) delle lettere distinte, presenti nella parola,
2) FREQ: contiene le frequenze con cui le lettere precedenti sono presenti nella parola,
3) IND1: contiene la posizione alfabetica delle lettere distinte, ordinate però secondo la loro posizione all'interno della parola;

4) FREQ1 contiene le frequenze associate ad IND1.

Questi vettori insieme ad altre due variabili:

I = lunghezza della parola;
K = numero di lettere distinte nella parola;

vengono restituiti al programma principale, dove viene costruito un vettore di "confronto", nel seguente modo:

I	K	IND	FREQ	PAROLA	IND1	FREQ1

Esso viene chiamato PARAG quando è costruito in base alla parola errata e VOCAB quando è costruito in base a quella potenzialmente esatta.

Esempio: se la parola è OGGI si ha:

IND(10) = 7, 9, 15
FREQ(10) = 2, 1, 1
IND1(10) = 15, 7, 9
FREQ1(10) = 1, 2, 1
I = 4
K = 3

3.3. Condizioni di accesso alle routine di correzione

Data una parola errata, non sappiamo quale tipo di errore è stato commesso. È importante notare che la stessa parola errata può essere generata da diversi tipi di errori.

Per esempio, data la parola OSA, essa può derivare da COSA, ROSA, OSSA, IOSA, per omissione di una lettera, oppure da ORA per lo scambio di un tasto.

Da questa osservazione emerge il fatto che la relazione tra parola errata e parola esatta non è sempre univoca.

È perciò importante poter trovare tutte le parole che, con maggiore probabilità, si avvicinano alla stringa errata, in modo che se la lista di parole alternative prodotta contiene più di una parola, sarà l'utente a selezionare quella giusta. Perciò la stringa errata deve poter essere analizzata da tutte le routine di correzione. Il soddisfacimento delle condizioni di accesso ad una o più di tali routine, ci dà l'indice di probabilità che quella stessa stringa possa essere stata generata commettendo errori diversi tra loro, non contemporaneamente.

Questo avviene perchè le condizioni di accesso alle varie routine di correzione sono necessarie ma non sufficienti. Cioè, se sono verificate determinate condizioni, è possibile che ci sia stato un errore di un certo tipo, ma non ne abbiamo la certezza. Solo dopo che la parola sarà stata analizzata dalla routine avremo un risultato certo.

Esaminiamo ora i vari tipi di errore previsti e le relative routine di ricerca e correzione.

3.4. Trasposizione di due caratteri adiacenti

Invertendo l'ordine posizionale di due caratteri adiacenti, si può dar luogo ad una stringa che non ha senso compiuto, oppure ad una parola con significato diverso da quello voluto. Ad esempio, la parola NANO può dare origine a NAON o anche ad ANNO.

L'algoritmo che scopre e corregge gli errori di trasposizione di due lettere adiacenti, si basa su un metodo che, essenzialmente, non tiene conto della posizione dei caratteri errati all'interno della parola.

Una parola in cui è stato commesso un errore di questo tipo ha evidentemente la stessa lunghezza della parola corretta, nonchè lo stesso numero di lettere distinte con la stessa frequenza.

Costruito il vettore di confronto PARAG, viene costruito il vettore VOCAB per la parola potenzialmente esatta.

Dopo di che vengono confrontati IND con IND' e FREQ con FREQ', che devono essere rispettivamente uguali, per quanto detto precedentemente. A questo punto viene fatto un confronto "carattere a carattere" tra: PARAG che contiene la parola errata e VOCAB che contiene la parola potenzialmente esatta.

Quando vengono trovati due caratteri diversi si fa un confronto incrociato; se tale confronto, nonchè tutti i rimanenti risultano positivi, la parola corretta è quella contenuta in VOCAB; altrimenti si prosegue nella ricerca di altri tipi di errore.

Un esempio chiarisce meglio la procedura. Si abbia la parola CSAA, che corrisponde a CASA con l'inversione della 2ª lettera con la 3ª. Costruiti i due vettori PARAG e VOCAB, si verifica che è IND = IND' e FREQ = FREQ'. Si eseguono allora i confronti sulle due parole, carattere per carattere. La 1ª lettera è uguale in entrambe. La 2ª è invece diversa; si confronta allora la 2ª e 3ª lettera di CSAA rispettivamente con la 3ª e la 2ª lettera di CASA; il confronto è positivo e quindi si prosegue. I rimanenti confronti sono tutti positivi, per cui si deduce che la parola contenuta in VOCAB è quella corretta.

3.5. Omissione di una lettera

Errore di omissione di una lettera significa ovviamente che è stato saltato un carattere della parola. Ad esempio, invece di CASA, si può avere: ASA; CSA; CAS; ecc.

Nel fare un tale errore, si può omettere un carattere che compare nella parola con frequenza 1, per cui esso non sarà più annoverato nel numero delle lettere distinte che compongono la parola; oppure si può omettere un carattere che compare con frequenza maggiore di 1, per cui esso verrà annoverato ancora nel numero di lettere distinte che compongono la parola. In entrambi i casi la frequenza relativa di quel carattere risulterà decrementata di 1 e la stringa in esame presenterà una lunghezza di una unità inferiore, rispetto a quella della parola potenzialmente esatta. Queste considerazioni sono alla base dell'algoritmo usato.

Se sono verificate le ipotesi precedenti, si procede al confronto delle parole nel modo seguente. Non appena vengono trovati due caratteri diversi, si mantiene fermo un indice, si incrementa di 1 l'altro e si continuano i confronti.

Ad esempio, date le parole CSA e CASA, poichè il confronto tra i due secondi caratteri è negativo, si confronta il 2° carattere di CSA col 3° di CASA, il 3° di CSA col 4° di CASA.

Se i rimanenti confronti non sono uguali, si esce dalla routine. Altrimenti (come nell'esempio fatto) si continua calcolando la differenza tra

le FREQ1 della parola potenzialmente corretta e quelle della parola errata. Se l'ipotesi di omissione di una lettera è vera, la somma di tali differenze deve essere uguale a 1, sia nel caso in cui è stata omessa una lettera con frequenza > 1, che nel caso in cui è stata omessa una lettera con frequenza = 1.

Un esempio del primo caso è dato da CASA e CSA:

Nella prima parola FREQ1 vale 1,2,1, mentre nella seconda vale 1,1,1; la somma delle differenze è quindi 1. Un esempio del secondo caso è dato ancora da CASA e da CAA (FREQ1 : 1,2,0).

La parola generata dopo un errore di omissione, sottoposta all'agoritmo di correzione, può dar luogo a più parole di senso compiuto. Per esempio, CSA può fornire oltre a CASA anche COSA.

Poichè la relazione tra parola errata e parola esatta non è univoca, viene in questo caso chiesto all'utente di selezionare quella che lui intendeva. Il determinare automaticamente se in quel determinato contesto si voleva intendere CASA oppure COSA esula dagli obiettivi di questo studio.

3.6. Raddoppiamento di una lettera

Questo errore si verifica quando inavvertitamente viene raddoppiato un carattere contenuto nella stringa; per esempio IEERI al posto di IERI. Si constata facilmente che una parola in cui è stato commesso un errore di raddoppiamento ha una lunghezza superiore di una unità rispetto alla parola potenzialmente esatta; il numero di lettere distinte è invece lo stesso mentre la frequenza relativa a quella lettera è incrementata di 1.

Per verificare queste premesse, l'algoritmo controlla che IND =IND', e che la somma (ICONT) delle differenze tra FREQ e FREQ' sia = 1. Dopo di che confronta le due parole, carattere a carattere. Quando vengono trovati due caratteri diversi, viene incrementato un contatore. Se esso diventa > 1, si esce dalla routine, perchè con un errore di raddoppiamento due caratteri al massimo possono essere diversi.

Altrimenti viene incrementato di 1 anche l'indice di PARAG; e si proseguono i confronti. Se questi risultano positivi, se il contatore è uguale ad 1 e se ICONT = 1, allora la parola corretta è quella contenuta in VOCAB.
Anche quando si commette un errore di raddoppiamento di una lettera si può dar luogo ad una parola di senso compiuto; ad esempio CASSA invece di CASA.

La parola CASSA viene considerata corretta ortograficamente, anche se in effetti è il frutto di un errore ortografico, perchè il programma non compie analisi semantiche.

3.7. Scambio tasto adiacente

Gli errori generati premendo un tasto adiacente a quello corretto possono essere di due tipi:

a) Un tasto viene scambiato con quello adiacente e nella parola in questione appare un carattere al posto di un altro. Ad esempio, essendo sulla tastiera i caratteri S e T contigui, si può scrivere CATA invece di CASA.

b) Oppure possono essere premuti contemporaneamente due tasti adiacenti, introducendo nella parola in questione un carattere in più. Ad esempio, essendo sulla tastiera i caratteri B e C contigui, si può scrivere CBASA invece di CASA.

Nella definizione dell'algoritmo di correzione occorre tener presente che lo scambio o l'introduzione di un carattere, può avvenire con un tasto adiacente a destra, oppure adiacente a sinistra. (Non consideriamo invece i due tasti adiacenti nelle righe immediatamente sopra e sotto il tasto corretto, perchè questo tipo di scambio è molto improbabile).

Allo scopo di avere il quadro delle adiacenze possibili su una tastiera, viene costruita una matrice di vicinanza (MAVIC), contenente i caratteri alfabetici, i caratteri speciali e le cifre da 0 a 9.

Nella prima colonna sono memorizzati i caratteri da considerare, nella seconda quelli adiacenti a destra, nella terza quelli adiacenti a sinistra.

Descriviamo sinteticamente l'algoritmo relativo al primo tipo di errore, cioè lo scambio di carattere (come CATA invece di CASA). Viene fatto un confronto, carattere a carattere, delle due parole; quando vengono trovati due caratteri diversi, si effettuano le seguenti operazioni: viene incrementato di 1 un contatore (ITOT), e vengono memorizzati i due caratteri diversi tra loro (S e T nell'esempio) e la posizione in cui sono stati trovati.

A questo punto viene consultata la matrice di vicinanza, cercando il carattere della parola potenzialmente corretta (S). Se questa si trova in una certa posizione (X, 1) della matrice, si va a vedere se l'altro carattere diverso (T) è uguale a (X, 2) oppure a (X, 3). Se non viene trovato, si esce dalla routine. In caso contrario, se i rimanenti confronti risultano positivi e se ITOT = 1 (ossia abbiamo trovato solo una volta due caratteri diversi), si conclude che la parola corretta è effettivamente quella contenuta in VOCAB.

Nel secondo caso, cioè inserimento di una lettera spuria (come CBASA invece di CASA), l'algoritmo è leggermente diverso, ma ugualmente basato sulla consultazione della matrice di vicinanza.

3.8. Intervallo di ricerca nel dizionario.

Il calcolo dell'intervallo entro cui effettuare le ricerche della parola corretta, viene fatto consultando la matrice LUPLED, i cui elementi, ordinati sulle righe secondo la lunghezza della parola e sulle colonne secondo il numero di lettere distinte, rappresentano gli indirizzi per accedere ad un certo gruppo di parole contenute nel dizionario.

Poichè il token errato può potenzialmente essere esaminato da tutte le routine di correzione, bisogna calcolare tutti gli intervalli entro cui effettuare le ricerche della parola esatta, in base alle modifiche della lunghezza e del numero di lettere distinte che un determinato errore apporterebbe alla parola corretta, e farne l'unione per prendere l'intervallo più grande.

Qualcuno di questi intervalli può essere vuoto; ciò dipende essenzial-

mente dalla struttura della lingua, perchè per esempio non si trovano mai tre consonanti consecutive, oppure certe combinazioni tra consonanti e vocali non sono permesse. Questo fa sì che il numero di lettere distinte all'interno di una parola sia limitato.

3.9. Non appartenenza al dizionario

Se la parola errata dopo essere stata analizzata da tutte le routine di cui soddisfa le condizioni d'accesso, ed essere stata confrontata con tutte le parole possibili in un certo intervallo del dizionario, non viene corretta, significa che la potenziale parola corretta non appartiene al dizionario a nostra disposizione. Di ciò, viene allora data segnalazione all'utente.

La condizione di non appartenenza al dizionario, viene verificata mediante un test sui "flag" che segnalano se l'effettuazione di una qualsiasi routine di correzione ha trovato una parola potenzialmente corretta.

Se tutti i flag sono uguali a zero, allora la parola risulta non appartenere al dizionario.

3.10. Andamento dei tempi di correzione

Per valutare la complessità di tempo relativa agli algoritmi usati, dobbiamo come prima cosa considerare i criteri di frazionamento del dizionario, in base ai quali si snelliscono i tempi di ricerca. Essi sono due: il primo permette un frazionamento in base alla lunghezza della parola da cercare e il secondo in base al numero di lettere distinte che costituiscono le parole contenute nell'intervallo precedentemente determinato.

Siano:

I = lunghezza parola
K = numero lettere distinte

$$\alpha_I = \frac{\text{tot. parole lunghe I}}{\text{tot. parole}}$$

$$\beta_{I,K} = \frac{\text{tot. parole lunghe I, con K lettere distinte}}{\text{tot. parole lunghe I}}$$

Se N è la cardinalità del dizionario, gli intervalli entro cui effettuare le ricerche sono dati dalla relazione:

$$N_{I,K} = \beta_{I,K}\ \alpha_I\ N$$

dove $N_{I,K}$ è il numero di parole di lunghezza I, aventi K lettere distinte.

Ogni routine di correzione utilizza almeno un $N_{I,K}$ relativo all'errore che deve correggere; infatti dato un token errato definito da una coppia di valori I, K si ha che:

1) la routine di *inversione* effettua le sue ricerche all'interno dell'intervallo $N_{I,K}$
2) la routine di *raddoppiamento* nell'intervallo $N_{I-1,K}$
3) la routine di *omissione* negli intervalli $N_{I+1,K}$ e $N_{I+1,K+1}$
4) la routine di scambio tasto adiacente negli intervalli $N_{I-1,K-1}$ e $N_{I,K}$

Detto N* il maggiore intervallo $N_{I,K}$, si ha, secondo dati sperimentali relativi al nostro campione di dati, che N* è sempre $<<$ N (vedi tabella fig. 4), per cui la complessità di tempo totale di questi algoritmi ha un andamento lineare.

4. Conclusioni

La "correzione ortografica" rappresenta il primo passo verso un tipo di correzione molto più complessa che è quella della struttura logica di una frase e delle relazioni tra i suoi componenti, in un linguaggio naturale.

Nel contempo rapresenta anche un punto di arrivo in quanto fornisce un valido aiuto nell'ottimizzazione dei tempi di produzione di documenti, poichè permette di eliminare una fase a volte lunga e noiosa quale quella della revisione manuale di un testo.

Numero di caratteri distinti (K)

Lunghezza della parloa (I)	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	1	1	0	0	0	0	0	0	0
4	0	6	101	266	0	0	0	0	0	0
5	0	1	79	403	557	0	0	0	0	0
6	0	0	20	244	791	585	0	0	0	0
7	0	0	5	148	756	1265	529	0	0	0
8	0	0	4	99	489	1307	1242	309	0	0
9	0	0	0	9	76	340	545	317	56	0
10	0	0	0	1	30	170	398	422	128	5

Fig. 4 - Analisi di un campione di 11.695 parole italiane di uso comune. Ogni casella della matrice fornisce il numero $N_{I,K}$ di parole aventi una lunghezza di I caratteri, K dei quali tra loro distinti.

5. Bibliografia

[1] GLANTZ H.T., *On the recognition of information with a digital computer;* J. ACM 4,2 (April 1957)

[2] BLAIR C.R., *A program for correcting spelling errors;* Inform. and Control 3, (March 1960)

[3] DAMERAU P.J., *A technique for computer detection and correction of spelling errors;* Comm. ACM 7,5 (March 1964), 171, 176

[4] HARMON L.D., *Automatic recognition of print and script;* Proc. IEEE 60, 10 (Oct. 1972)

[5] Mc ELWAIN C.K. and EVANS M.E., *The degaroler - a program for correcting machine - read Morse code;* Inform. and Control 5, 4 (Dec. 1962)

[6] VOSSLER, C.M. and BRAUSTON, N.M., *The use of context for correcting garbled English Text;* Proc. ACM-Nation. Convention, Aug 1964

[7] CARLSON G., *Techniques for replacing characters that are garbled on input;* Proc. 1966 Spring Joint Comp. Conf., AFIPS Press, Arlington, Va.

[8] CARNEW, R.W., *A statistical method of spelling correction;* Inform. and Control 12,2 (Feb. 1968)

[9] RISEMAN, E.M. and EHRICH, R.W., *Contextual word recognition using binary diagrams;* IEEE Trans. Comp. C-20, 4 (April 1971)

[10] RISEMAN, E.M. and HAUSON A.R., *A Contextual post processing system for error correction using binary n-grams;* IEEE Trans. Comp. C -23,5 (May 1974)

[11] MORRIS R. and CHERRY L.L., *Computer detection of typographical errors;* IEEE Trans. Profess. Comm. PC 18, 1 (march 1975)

[12] Mc MAHON et al., *Statistical text processing;* Bell Syst. Tech J. 57,6 part 2 (July-Aug. 1978)

[13] J.L. PETERSON, *Computer Programs for detecting and correcting spelling errors;* Computing Practices

[14] J.L. PETERSON, *Design of a spelling program: An experiment in program design;* Lecture Notes on Compt. Science, 96, Spring-Verlag, N.Y, (Oct. 1980)

[15] WAGNER, R.A., and FISCHER, M.S., *The string-to-string correction problem;* J. ACM 21, 1 (Jan. 1974)

[16] MORGAN, HL., *Spelling correction in systems programs;* Comm. ACM 13,2 (Feb. 1970)

[17] LOWRANCE, R., and WAGNER, R.A., *An extension of the string-to-string correction problem;* J. ACM (22,2 (April 1975)

[18] WONG C.K., and CHANDRA, A.K., *Baunds for the string editing problem;* J.ACM 23,1 (Jan 1976)

[19] TERUO OKUDA, EIICHI TANAKA, TAMOTSU KASAI, *A method for the correction of garbled words based on the Levenshtein metric;* IEEE Trans. on Computer, C 25, 2 (Feb. 1976)

[20] MILLER, L. *EPISTLE: a system for the automatic analysis of business correspondence;* Proceedings of National Conference on Artificial Intelligence, Stanford University (Aug. 1980)

[21] ZAMPOLLI R., Lessico di frequenza della lingua italiana.

Criteri per la progettazione del software didattico

GIANNA DOTTI MARTINENGO

DIDA. EL
Milano

1. Introduzione

Introdurre innovazioni in un sistema educativo richiede un processo lento ed incerto. La Computer Based Education (CBE) è la più recente tra le nuove tecnologie didattiche a doversi confrontare con questa realtà.

Le fasi che formatori e docenti possono attraversare quando devono confrontarsi con una nuova tecnologia di solito assumono questa articolazione (fig. 1):

- *Presa di coscienza del problema*
- *Interesse*
- *Esame critico*
- *Utilizzo in fase sperimentale*
- *Adozione*

Gli stessi atteggiamenti vengono assunti quando le nuove tecnologie investono settori come l'intrattenimento e le attività di lavoro. Ma mentre il processo in questi casi ha andamenti assai rapidi, l'innovazione nell'ambito dell'istruzione segue un corso di adattamento molto rallentato. Le cause di questa lentezza possono essere individuate in:

- risorse e finanziamenti limitati
- scarsi investimenti nella ricerca e nello sviluppo
- difficoltà di misurazione dei risultati e di valutazione delle innovazioni in termini di efficacia.

Se analizziamo gli ultimi 10 anni di CBE, possiamo rilevare quanto sia stata rapida l'evoluzione che ha portato ad una significativa riduzione dei costi e ad una sempre crescente presa di coscienza dell'efficacia di queste nuove tecnologie nel processo di insegnamento/apprendimento.

Numerosi studi di valutazione sono stati finanziati allo scopo di verificare in quale misura il CBE, confrontato con i sistemi tradizionali, incida sulle tecniche di insegnamento e sui processi di apprendimento. [1,2,3,4].

I risultati sono stati positivi. L'utilizzo del CBE ha portato in tutti i livelli di formazione a (fig. 2):

- *Miglioramento in termini di apprendimento, sia sotto il profilo cognitivo che per gli aspetti motivazionale e relazionale.*
- *Riduzione dei tempi di istruzione* (tempo studenti, tempo insegnanti)
- *Aumento del numero degli studenti che possono accedere all'istruzione senza variare il numero dei docenti e la quantità delle ore di insegnamento.*
- *Espansione dei curricoli* (le potenzialità offerte dall'elaboratore consentono l'accesso a settori di alcune discipline prima difficilmente trasferibili con i sistemi tradizionali).
- *Raggiungimento di alcuni obiettivi fondamentali del processo educativo, quali l'individualizzazione dell'insegnamento.* Il CBE infatti rende possibile per la prima volta:
  - la rilevazione dei livelli di partenza di ogni allievo,
  - la definizione di obiettivi didattici secondo la rilevazione precedente,
  - la verifica sistematica del ritmo e dell'itinerario di apprendimento di ciascuno,
  - l'effettuazione di interventi di sostegno secondo una strategia individualizzata.
- *Riduzione dei tempi necessari per le operazioni di raccolta dati e misurazione.*
- Possibilità di condurre gli allievi *al raggiungimento degli stessi livelli di conoscenza* attraverso percorsi e tempi individualizzati.
- *Rapporto equilibrato tra costi e benefici.*

L'aumento del bisogno di istruzione permanente, l'aumento dei costi di formazione attraverso sistemi tradizionali, la diminuzione dei costi delle tecnologie che forniscono istruzione, fanno ipotizzare ampie prospettive di affermazione dell'uso di sistemi multimediali nelle attività di insegnamento.

2. Rapporto tra responsabile della formazione e nuove tecnologie

Qual'è l'atteggiamento che il responsabile della formazione deve assumere di fronte alle Nuove Tecnologie di Istruzione? (Fig. 3).

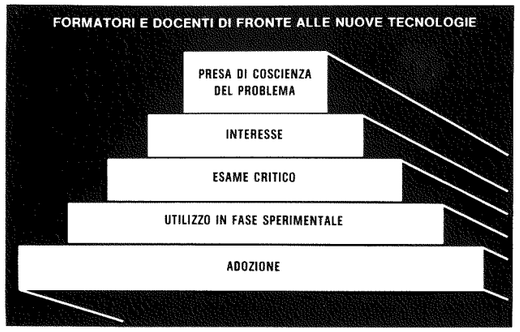


Fig. 1

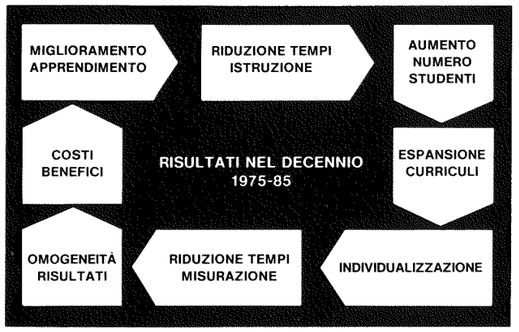


Fig. 2

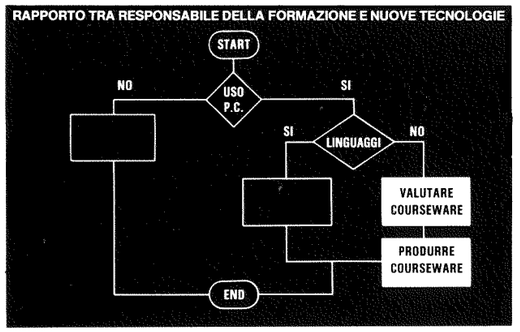


Fig. 3

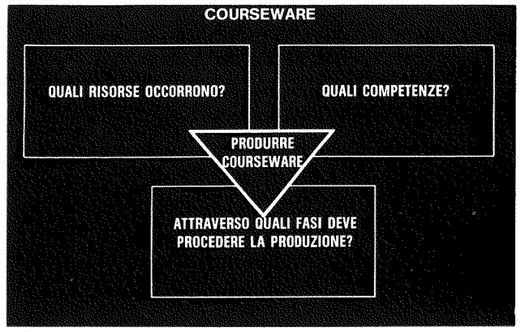


Fig. 4

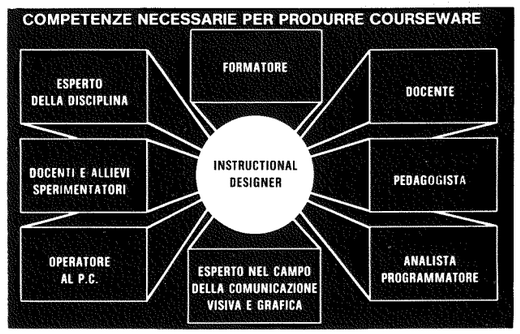


Fig. 5

Fig. 6

Il formatore di fronte al CBE (Computer Based Education): l'approccio generale (fig. 1); i risultati ottenibili (fig. 2); le aree di intervento del formatore (fig. 3); le valutazioni preliminari (fig. 4) e le competenze necessarie in un team di progetto courseware (fig. 5); le fasi in cui si articola un progetto CBE (fig. 6).

- Rifiutare l'utilizzo dell'elaboratore?
- Padroneggiare le procedure meccaniche d'uso?
- Imparare i linguaggi di programmazione?
- Imparare a valutare il courseware (software didattico)?
- Produrre il courseware?

Dando come presupposto che il rifiuto dell'elaboratore è smentito di fatto dalla realtà quotidiana, si ritiene indispensabile per il formatore sapere utilizzare manualmente lo strumento. Difficile e non produttivo risulta l'investimento di energie, da parte di chi ha già una professionalità in un settore diverso, nell'acquisizione dei linguaggi di programmazione. Indispensabile è invece pervenire a valutare in modo critico e secondo parametri precisi la qualità intrinseca del courseware, la sua corrispondenza alle finalità a cui è destinato e il rapporto costo/beneficio.

Per chi è intenzionato a far parte di un gruppo di lavoro che voglia progettare e produrre courseware è essenziale conoscere (fig. 4):

a) *quali risorse occorrono*

b) *quali competenze*

c) *attraverso quali fasi deve procedere la produzione.*

Queste conoscenze sono indispensabili anche per chi intenda valutare il courseware. Occorre infatti che egli sia in grado di controllare tutti gli aspetti della produzione, dalla analisi preliminare alla documentazione, per non dare un giudizio sommario del prodotto. Gli investimenti richiesti dal CBE sono notevoli, pertanto si richiede che i risultati in termini di efficacia didattica siano significativi.

Per raggiungere questo obiettivo è necessario che il courseware abbia un livello qualitativo elevato e pertanto sia frutto di lavoro interdisciplinare di specialisti ad alto livello di professionalità.

a) *Risorse per produrre courseware*

Sono state effettuate stime del numero di ore necessarie per produrre un'ora/studente. Raramente queste stime sono concordi: si va da valori minimi di circa 30 ore a massimi di 400 ore/autore per ora/studente. In realtà una semplice analisi rivela che questi valori non sono confrontabili se non corrispondono a precise valutazioni delle fasi di sviluppo del courseware e del tipo di prodotto che si vuole ottenere (livelli di interattività, grado di individualizzazione, uso di grafica, etc.).

b) *Competenze necessarie per produrre courseware*

Nel team di progetto devono essere presenti le componenti elencate in fig. 5. Di importanza centrale è la figura del capo progetto ("instructional designer"), cioè la persona che, con la collaborazione degli esperti dei singoli settori, imposta il progetto nella sua globalità e ne coordina tutte le fasi.

c) *Fasi in cui si articola un progetto*

Si possono distinguere le seguenti fasi (fig. 6):

- Analisi preliminare
- Sviluppo
- Implementazione
- Sperimentazione
- Documentazione

La successione delle fasi è solo apparentemente sequenziale, in realtà ogni momento retroagisce sul precedente e fa parte di una struttura organica che comprende tutte le fasi.

Nel seguito vengono dettagliati gli aspetti rilevanti delle singole fasi.

3. Fasi di un progetto

3.1. Analisi preliminare

In questa fase iniziale, il gruppo di lavoro coordinato dall'instructional designer compie la serie di operazioni illustrata in fig. 7, che termina con le strategie di intervento. Queste sono definite schematicamente in fig. 8.

3.2. Lo sviluppo

Questa fase può essere considerata uno sviluppo analitico della precedente e si articola in diverse voci (fig. 9), che vengono qui di seguito illustrate singolarmente.

3.2.1. *Interazione*

Con questo termine si intende l'insieme delle scelte che il gruppo di lavoro compie relativamente ad una serie di argomenti (fig. 10).

a) *Percorsi*

Il controllo sull'itinerario di apprendimento può essere gestito o *dall'insegnante* o *dall'allievo* o *dal programma.*

Nel 1° caso *(teacher-controlled)*, il docente fissa, a seconda degli obiettivi relativi ad ogni studente:

- la durata della sessione
- il tempo di risposta
- il numero dei tentativi concessi
- la soglia percentuale (mastery) che definisce il livello di competenza ritenuto necessario per passare ad una difficoltà superiore o inferiore.

Nel 2° caso *(learner-controlled)* spetta all'allievo decidere il proprio percorso di apprendimento (scelta dell'attività che intende svolgere, del numero delle esercitazioni, del livello di difficoltà, del ritmo di esecuzione, ecc.

Nel caso in cui il programma guidi il percorso *(program-controlled)*, le operazioni precedenti vengono regolate automaticamente.

b) *Movimento logico*

I percorsi che si possono disegnare in un *program-controlled* CBE possono essere: lineare, ramificato, ad alto livello di individualizzazione (fig. 11).

- Il disegno del programma è *lineare* quando lo studente procede attraverso una sequenza fissa di esercizi o pagine di istruzione indipendentemente dalla sua "performance"
- Il disegno del programma è *ramificato* quando il programma valuta come lo studente sta procedendo e automaticamente presenta o omette parti del programma sulla base delle risposte dell'allievo.
- il disegno è *ad alto livello di individualizzazione* (AIS = Adaptive Instructional System) quando il programma predispone percorsi individualizzati che si differenziano per qualità, gradualità di eser-

Fig. 7

Fig. 8

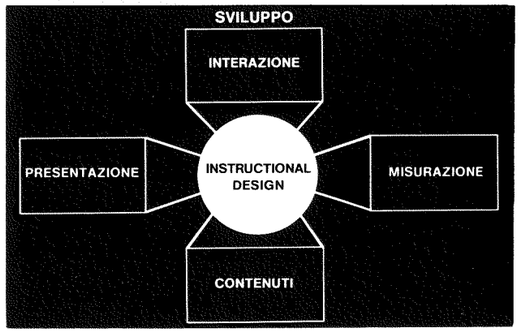


Fig. 9

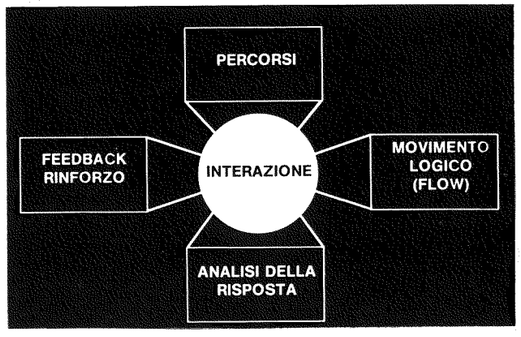


Fig. 10

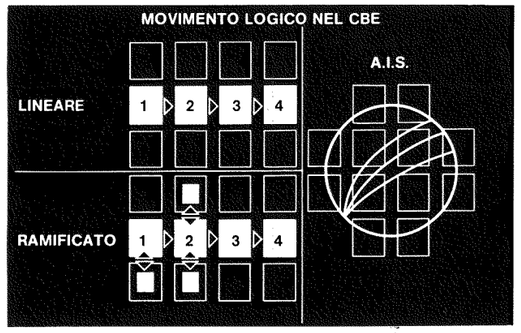


Fig. 11

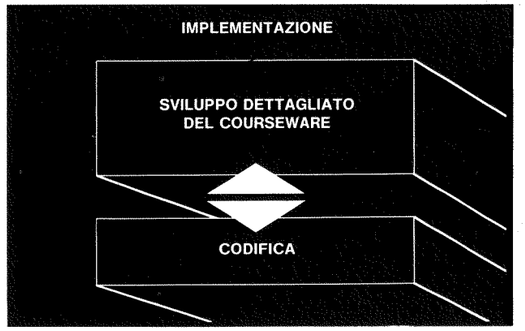


Fig. 12

Fig. 13

Le fasi di un progetto courseware:
analisi preliminare (fig. 7,8);
sviluppo (fig. 9, 10, 11);
implementazione (fig. 12);
sperimentazione (fig. 13).

cizi e tempi di percorrenza, *adattandosi* continuamente allo stile di apprendimento di ogni studente.

Questo è reso possibile dai criteri decisionali (motion algorithms) formulati dal programma sulla base del tipo di risposta e modalità di risposta (tempo, numero tentativi, numero delle performance corrette o errate, numero delle performance consecutivamente corrette o errate, etc.).

c) *Analisi della risposta*

L'analisi della risposta da parte del programma può avere diversi criteri di gradualità in funzione degli obiettivi.

Vengono distinti gli errori meccanici dovuti a:

- errata digitazione
- non comprensione delle istruzioni
- mancata esecuzione per fuori tempo
- tentativo da parte dello studente di non eseguire il compito richiesto

dagli errori logici o di contenuto: in questo caso la gradualità può essere di questo tipo: il programma si limita ad indicare se la risposta è giusta o errata, oppure utilizza l'occasione per fornire supplementi d'informazione (giusto perchè...., sbagliato perchè....)

d) *Feedback-Rinforzo*

La scelta del *feedback* deve essere guidata da opportuni criteri, e cioè i commenti siano:

- chiari, comprensibili, adeguati al livello dei destinatari
- pertinenti alle risposte
- corretti dal punto di vista formale e del contenuto

Essi possono essere:

- usati solo dopo la risposta corretta, o dopo la risposta errata, o dopo entrambe.
- immediati o posticipati

In caso di risposta errata il rinforzo segnala la corretta esecuzione.

Il *rinforzo* può essere:

- grafico (disegno, asterisco, flash, reverse, etc.)
- sonoro (beep, musichetta, etc.)
- testo (corretto, esatto, etc.)

Il tipo di rinforzo scelto non deve essere distraente.

3.2.2. *Presentazione*

Anche la "presentazione" della pagina video ha una funzione didattica, pertanto deve rispondere ad una serie di criteri:

- siano evitate soluzioni grafiche e sonore che possono distrarre dall'obiettivo del programma
- la grafica sia utilizzata in modo appropriato dal punto di vista didattico
- i caratteri siano chiari e ben spaziati
- vengano usati segnali grafici per attirare l'attenzione (flash, reverse, sottolineature, caratteri maiuscoli/minuscoli) e la scelta dei segnali grafici sia coerente in tutto il programma
- il colore e il suono siano finalizzati agli obiettivi didattici
- i diversi tipi di testo di una pagina video (titolo, istruzioni, enunciato, risposta, commento) siano chiaramente evidenziati e distinti
- il livello linguistico dei vari tipi di testo (titolo, istruzioni, messaggi, commenti) sia formalmente soddisfacente e adeguato ai destinatari
- le consegne per l'esecuzione del compito siano chiare, precise ed univoche, formalmente corrette e adeguate al destinatario.

3.2.3. *Contenuti*

I contenuti sono selezionati in funzione degli obiettivi fissati nel progetto formativo.

Esercizi, problemi, domande sono presentati allo studente perchè possa esercitare le abilità richieste.

Il materiale presentato deve essere coerente con le finalità didattiche e gli obiettivi dichiarati.

Lezioni/unità/attività siano organizzate secondo livelli di gradualità.

3.2.4. *Misurazione*

Per attualizzare la misurazione, il progetto può prevedere:

- test di ingresso a scopi diagnostici
- un test finale per la valutazione
- rapporto di sessione per la rilevazione delle performance

La misurazione può essere espressa o in percentuale, o con un commento/voto o con un rapporto statistico dettagliato individuale o per classe/gruppo.

Il rapporto statistico può essere memorizzato su disco personale o su disco gruppo/classe.

L'allievo e/o il docente possono accedere alla misurazione:

- a fine sessione
- in ogni momento della sessione.

3.3. Implementazione

Tutti i criteri precedentemente citati vengono trasformati in unità didattiche concretizzate in "frame" con la tecnica dello storyboard.

Dopo questa fase si inizia la *codifica.*

Il programmatore, attraverso un linguaggio di programmazione o un sistema autore, traduce il progetto in un programma su elaboratore.

Nella fase di analisi preliminare erano stati confrontati i due strumenti (linguaggio di programmazione o sistema autore) per individuare quello più funzionale al progetto.

Le caratteristiche dei linguaggi di programmazione sono:

- maggiore flessibilità
- basso costo
- tempi lunghi di realizzazione
- non completa affidabilità
- tutto deve essere programmato
- linguaggio difficile da apprendere

Il sistema autore invece presenta:

- minore flessibilità
- alti costi
- tempi più brevi di realizzazione
- maggiore affidabilità
- non tutto deve essere programmato
- linguaggio più facile da imparare.

3.4. Sperimentazione

Il courseware una volta realizzato deve essere verificato attraverso la sperimentazione da parte di uno o più gruppi pilota al fine di raccogliere le informazioni necessarie per gli eventuali aggiustamenti (fig. 13).

Dopo la revisione il courseware deve essere documentato e successivamente pubblicato e distribuito.

3.5. Documentazione

- La documentazione deve essere chiara e completa
- Le istruzioni siano corrette (senza discrepanze tra le note operative e l'effettivo funzionamento del programma)
- Occorre prevedere una guida didattica per l'insegnante che includa:
  - la definizione dei modelli teorici di riferimento
  - la definizione delle finalità
  - la definizione del pubblico cui è indirizzato il corso
  - la definizione degli obiettivi
  - la presentazione dei contenuti
  - l'indicazione dei criteri di scelta dei contenuti
  - la struttura del corso/programma
  - la presentazione dei tipi di esercizi
  - la modalità di esecuzione
  - criteri per individuare i livelli di partenza
  - un esempio di funzionamento
  - i suggerimenti per attività integrative e collaterali
  - criteri per la misurazione sistematica
  - una bibliografia

4. Bibliografia

[1] KULIK, C.C., KULIK, J.A., & BANGERT-DOWNS, R.L. *Effects of computer-based education on elementary school pupils.* A symposium paper presented at the annual meeting of the American Educational Research Association, New Orleans. (April 1984).

[2] KULIK, J.A., KULIK, C.C., & COHEN, P.A. *Effectiveness of the computer-based college teaching:* A meta analysis of findings, Review of Educational Research, 50, 525-44 (Dec. 1980).

[3] KULIK, J.A., BANGERT-DOWNS, R.L., & WILLIMS, G.W. *Effects of computer-based teaching on secondary school students.* Journal of Educational Psychology, 75, 19-26 (1983).

[4] KULIK, J.A., & BANGERT-DOWNS, R.L. *Effectiveness of technology in precollege mathematics and science teaching.* Journal of Educational Technology Systems, 12, 137-58 (1984).

Collana dei Quaderni di Informatica

La collana di testi "Quaderni di Informatica" rappresenta la proiezione sul piano librario della presente rivista. La direzione scientifica della collana (che è edita da F. Angeli) è del "Centro Studi Informatica e Automazione" della Honeywell I.S.I. A titolo di rassegna sintetica, riproduciamo qui di seguito le copertine dei volumi finora usciti, che coprono alcune delle più importanti aree dell'informatica.