UNIVERSITÀ DI BRESCIA
FACOLTÀ DI INGEGNERIA
Dipartimento di Elettronica per l'Automazione

Laboratorio di Robotica Avanzata
Advanced Robotics Laboratory

Corso di Robotica
(Prof. Riccardo Cassinis)

Valutazione del sistema ARC per CognaChrome

Elaborato di esame di: **Emanuele Gnali**

Consegnato il: **20 marzo 2001**

Sommario

Il Laboratorio Didattico di Robotica Avanzata della facoltà utilizza proficuamente il sistema di visione CognaChrome, applicato ad un robot Pioneer I, per lo svolgimento di attività didattiche e sperimentazioni su robot autonomi. In particolare il sistema di visione è gestito da un sistema di controllo e sviluppo denominato mini-ARC che, nella versione in possesso, consente di interfacciare con naturalezza CognaChrome a Saphira, l'ambiente di controllo e programmazione ad alto livello del robot Pioneer. Scopo della presente attività è stata la valutazione dell'opportunità di acquistare la versione completa del sistema ARC.

1. Introduzione

Il sistema di visione CognaChrome è il connubio tra hardware specializzato nell'acquisizione di immagini da videocamera, costituito da un microcontroller ed una sezione di interfaccia A/D, e software orientato principalmente all'elaborazione delle immagini acquisite.

In particolare il software, complessivamente denominato ARC, si divide su due piattaforme:

- il "Target Software" è la porzione eseguita dal microcontroller: consiste in un sistema operativo (ARC Kernel), in grado di eseguire in real-time il codice per l'elaborazione delle immagini, per la comunicazione con il mondo esterno (tipicamente il sistema "HOST" via seriale), ecc. Il codice che arriva pre-programmato col sistema permette il riconoscimento a 60fps di macchie colorate (BLOB) in tre colori diversi ("canali"), impostati attraverso una fase di addestramento, e la comunicazione con il sistema "HOST", tramite un semplice protocollo seriale, dei dati generati dal rilevamento e dei comandi per il sistema.
- l' "Host Software" gira invece su un PC collegato via seriale al CognaChrome, ed è un sistema interattivo che, oltre a permettere il comando diretto del CognaChrome stesso, si interfaccia direttamente a Saphira (software di controllo dei robot Pioneer), esportando in questo strutture ad alto livello per lo sfruttamento dei dati ottenuti dalla videocamera per il controllo del robot.

Il sistema CognaChrome in dotazione al Laboratorio Didattico di Robotica Avanzata della facoltà è pienamente funzionante ed utilizzabile, e molte esercitazioni si sono potute effettuare basandosi sulle sue capacità di inseguimento dei BLOB. L'interfacciamento diretto del software con Saphira ha reso infatti molto semplice incorporare nei "Behavior" e nelle "Activity"dei robot le informazioni visive assieme a quelle già contribuite dai sensori standard (un'idea abbastanza precisa di cosa si possa ottenere attualmente dal sistema CognaChrome la si può trovare nella "Cognachrome Vision System User's Guide").

Il passo successivo sarebbe l'estensione delle routine di elaborazione delle immagini acquisite al di là del semplice riconoscimento dei BLOB: l'ARC (Kernel ed Host Software) è in grado di effettuare l'upload di nuovo codice nella EPROM della scheda per estenderne le capacità, e questo è già consentito dalla versione ridotta di ARC (chiamata Mini-ARC), fornita di serie col sistema. Che invece manca, ed è venduta

separatamente come opzione dalla ditta produttrice del sistema CognaChrome stesso, è la possibilità di sviluppare proficuamente il nuovo codice.

Il sistema di sviluppo completo (ARC Development System), infatti, rende concretamente possibili la scrittura in C (ARC C), la compilazione ed il debug di tale software, attraverso librerie ad alto livello e strumenti di debug capaci di monitorare e guidare il codice mentre gira direttamente nel microcontroller della scheda.

2. Il problema affrontato

Le pochissime informazioni messe a disposizione dal sito della NewtonLabs (produttrice di CognaChrome e del relativo sistema ARC) sono assolutamente insufficienti a farsi un'idea di ciò che viene effettivamente offerto dal sistema ARC completo rispetto a mini-ARC. Il tutto viene infatti riassunto con un lapidario:

> *"When enabled with a Mini-ARC license key, this software allows you to download new software to your processor board, and interact with it over a serial port. When enabled with a full ARC Development System license, it also allows you to interactively view and change global variables, call functions, and debug programs running on the board."*

In pratica riduce l'acquisto della licenza a quello di un debugger che lavori attraverso il cavo seriale…

Si è cercato il contatto con la Ditta tramite e-mail, senza sortire alcun effetto alcuno. Nell'attesa di risposta, si è intanto provveduto a ripristinare l'installazione di tutto il software necessario al sistema di visione sui PC del laboratorio, appena passati per l'aggiornamento dei rispettivi sistemi operativi.

3. L'installazione di ARC

Innanzitutto si è scaricata l'ultima versione di ARC, la 1.4.1 per Linux, in formato eseguibile ELF – il vecchio formato "a.out" non è più supportato dalla versione 7 di Red Hat Linux, attualmente in funzione sui PC. Anche in questo caso il sito della NewtonLabs non è stato di grande aiuto, in quanto i link per il download non erano attivi… Fortunatamente una copia del software era messa a disposizione anche dal sito di ActivMedia Robotics, produttore del robot Pioneer, che commercializza il sistema CognaChrome con il nome "Fast Track".

Verificato quindi il corretto funzionamento di Saphira, è stato installato ARC che, dopo piccole difficoltà tecniche (per la cui soluzione si rimanda al paragrafo successivo), si è potuto verificare il corretto funzionamento di tutto il sistema.

Un'ulteriore amara sorpresa è giunta durante l'installazione: mentre sembrava che l'acquisto della "full license" semplicemente sbloccasse il funzionamento del debugger e la possibilità d'uso delle librerie, si è costatata invece la totale assenza di queste ultime. Nel pacchetto in nostro possesso ("ARC Serial Interaction Program") non erano inclusi neppure gli header, utili per farsi almeno un'idea di cosa potranno offrire le librerie. Queste, infatti, sono contenute in un altro pacchetto, "ARC Compilation Tools", che però, oltre ad essere distribuito solo nel vecchio formato "a.out" (incompatibile ormai con i PC del laboratorio), non era – di nuovo! – scaricabile dal sito. Si sono dovute quindi nuovamente attendere notizie dalla NewtonLabs prima di poter avere un'idea, anche superficiale, di quali porte si potranno aprire…

3.1. Note tecniche

Con la reinstallazione dei sistemi operativi sui PC del laboratorio si è reso necessario reinstallare anche ARC, operazione che non è stata esente da difficoltà di ordine tecnico.

Innanzitutto la versione precedentemente usata non è più compatibile con le nuove distribuzioni Linux (come la Red Hat 7 in uso adesso), in quanto queste sono interamente compilate (e richiedono che lo siano anche gli altri programmi) nel formato modulare ELF, che va a sostituire il più "monolitico" vecchio formato a.out.

Quindi, prelevata e posta in sede la versione ELF del pacchetto, si è reso necessario apportare alcune modifiche alla configurazione degli utenti dei PC affinché tutti gli interessati potessero godere dei diritti sulle risorse utilizzate da ARC – in particolare: il path ai suoi binari, la seriale collegata al robot via radio-modem (peraltro necessaria anche a Saphira), ed anche l'input del frame-grabber, in modo che a fianco della finestra di ARC si potesse visualizzare anche l'output reale della videocamera. A tal fine si è creato un gruppo utenti, denominato "", cui è stato garantito l'accesso a tali risorse.

Infine, perché la configurazione del path di esecuzione e degli alias di avvio per ARC fosse facilmente ripristinabile e portabile, essa non è stata inserita brutalmente nella configurazione di boot, ma è stata messa in un semplice programma batch che, posizionato in una apposita directory (/etc/profile.d), viene eseguito al boot. Il batch si chiama arc.sh, e crea gli alias:

arcp	equivalente al comando:	arc –pioneer
arcs	equivalente al comando:	arc –saphira

3.1.1. arc.sh

```
ARC=/usr/local/arc
export PATH=$PATH:$ARC/linuxbin

alias=arcs"arc -saphira"
alias=arcp"arc -pioneer"
```

4. Le informazioni raccolte

Evidentemente NewtonLabs sta proprio in questo periodo riordinando il proprio sito: alcuni indirizzi e link utilizzati all'inizio di questo lavoro sono scomparsi, mentre ripercorrendo il nuovo albero delle pagine ci si è finalmente imbattuti in qualcosa di scaricabile!

Abbiamo ora a disposizione tutti i pacchetti cercati fin dall'inizio, ma purtroppo non svelano granché ai nostri fini:

- **arc_host_linuxelf_arc_1.4.1.tar.z,** contiene il solo mini-ARC, nel formato eseguibile ELF;
- **arc_host_linux_arc_1.4.1.tar.z,** contiene ARC completo in formato a.out; in effetti, l'unica differenza (a parte il formato) con l'altro pacchetto per Linux è la presenza del programma TTLOAD, necessario al caricamento di sotware nella EPROM del microcontroller del CognaChrome;

- **arc_host_linux_tools_1.4.tar.z,** denominato "Compilation tools for Linux" (sempre nell'inutilizzabile formato a.out); anche questo non mostra header o sorgenti, solo eseguibili di supporto e codice oggetto di librerie per la compilazione di sorgenti C in codice per MCU 68332 compatibile con il sistema operativo ARC Kernel;
- **arc_host_win32_1.5b5.exe**, denominato "ARC+MiniARC v1.5 beta5 for Windows 95/NT", è il pacchetto più completo, comprendente sia ARC, sia tutti i tool di compilazione analoghi ai precedenti, completi in questo caso anche degli header documentati delle librerie fornite.

Il pacchetto per Windows è quindi l'unico in parte informativo; peccato che header e librerie forniti sono relativi al solo ARC Kernel, sistema operativo della scheda CognaChrome.

4.1. CognaChrome ARC Kernel Libraries

In effetti, il codice per la compilazione con l'ARC Kernel è molto utile alla comprensione del sistema – tutti gli header sono corredati da una sintetica documentazione che fa luce sull'uso che il Kernel fa della TPU e degli Interrupt del microcontroller, nonché sulla ben congegnata gestione di multitasking ed I/O che offre – e sicuramente sarà indispensabile durante l'ingegnerizzazione di eventuali grossi progetti che scendano a fondo nel sistema. Poco offrono invece alle prime avventure di programmazione della scheda, assolutamente zero per quanto riguarda l'implementazione di routine visuali. Bisogna ancora cercare altrove.

In particolare vengono offerte funzioni per la gestione a livello piuttosto alto dei processi [multi.h] e della seriale (gestibile sia come stdin & stdout stream, sia come file) [serial.h, tpu_uart.h]. Seguono le tipiche funzionalità ANSI C [kernel_malloc.h, kernel_stdio.h, ecc.], nonché uno strato HAL per la gestione dei dettagli della TPU e dell'hardware [tpu.h, tpuhigh.c, eeprom_332.h, config_332.h, interrupt_332.h]. A ciò si affiancano una console di comando unix/like (la linea di comando presentata dall-ARC Serial Interaction Program) [console.h, cmdline.h], il supporto al debug on-board [rpc.h] ed un set di librerie alternative per la compilazione dei programmi utente su PC [*_386.h anziché *_332.h].

I dettagli relativi alle varie funzioni del Kernel, molto ben documentate, si possono trovare nell' "ARC Development System User's Guide" [arc.pdf].

4.2. CognaChrome Vision Libraries

Dal sito della NewtonLabs, nell'area dedicata al sistema CognaChrome (meno specifica della sezione ARC), si può prelevare invece ciò che completa il firmware della scheda: le CognaChrome Vision Libraries. Nel microcontroller, infatti, sul già descritto ARC Kernel si possono eseguire programmi utente; in mancanza di essi (e come default all'acquisto del sistema CognaChrome), il sistema offre questo componente per il riconoscimento dei BLOB, ecc.

Nell'archivio **cognachrome_libs_27_0.tar.z** si possono trovare header e librerie per il collegamento di programmi utente al sistema di visione on-board. Purtroppo anche questi header sono molto parchi di informazioni: in particolare si nota un'interfaccia di altissimo livello al sistema, che non sembra permettere il riferimento ad entità più elementari di Frame e BLOB.

In particolare si trovano funzioni relative a: training e riconoscimento dei BLOB [blob.h, colorepts.h], aggiustamento di attributi dell'immagine (luminosità, ecc.), riconoscimento

di soglie e contorni relative ai BLOB [frameproc.h, linefind.h], generazione dei frame di output [colorplot.h, visdraw.pro]

A queste si affiancano funzioni di più basso livello: alcune necessarie solo al processing delle stesse entità, per esempio per l'avvio sincronizzato dell'elaborazione dei frame o di altri processi [frame.h, intvision.h, visdraw.pro]; vi sono funzioni di comunicazione [protocol.h, visui.h]; altre sono funzioni matematiche generiche [intmath.h, qsort.h]; altre ancora si occupano di aspetti più prettamente hardware, come la gestione del grabber [vstream.h], della EEPROM [eep.h], del bus I_2C [i2c.h], dell'altoparlante (!) [beep.h], nonché di eventuali motori collegati alle porte di I/O [servo.h].

Una nota: nel sistema CognaChrome in nostro possesso, specificatamente adattato per l'interfacciamento al robot Pioneer, è caricata un'altra versione delle librerie di visione, da usarsi "as-is" con Saphira: è liberamente scaricabile dal sito, in forma di ROM Image, per poter ripristinare il sistema alle condizioni iniziali. Il caricamento nel microcontrollore di altre librerie, più generiche, ne impedisce la comunicazione con Saphira. Lo stesso dicasi per i programmi utente, se non sono esplicitamente previste routine di comunicazione seriale con Saphira – in effetti nelle release-notes dell'ultima versione disponibile delle Vision Libraries ne viene descritta una drastica riduzione dell'occupazione di memoria, per cui i programmi caricati nello spazio residuo possono usufruire delle routine di comunicazione seriale già in uso.

4.3. Gli esempi

A fianco dell'archivio delle CognaChrome Vision Libraries si possono trovare anche un paio di esempi applicativi delle librerie stesse, molto istruttivi anche se non documentati, in quanto abbastanza semplici (archivio **cognachrome_examples.tar.z**):

- **ScampVis**: a giudicare dal nome e dalle funzioni è ispirato dal progetto SCAMP (Supplemental Camera and Maneuvering Platform) dell'Università del Maryland: una telecamera robotizzata subacquea che mantiene l'inquadratura dell'obiettivo fornendo un altro occhio agli operatori impegnati con altri robot. Nel programma presentato, delle 4 funzioni previste dal progetto completo (Following – inseguimento del target con posizione relativa ad esso fissa, Tracking – inseguimento visivo dell'obiettivo senza spostamento dall'area di lavoro, Surveying – osservazione dell'obiettivo da diversi punti di vista, e Moving-to – approccio all'obiettivo), le prime due sono implementate e selezionabili dalla console, tramite l'attivazione o disattivazione della funzione di traslazione. La simulazione effettuata dal programma restituisce le 6 coordinate di rotazione e traslazione per l'inseguimento del target, utilizzabili da un eventuale sistema di movimento reale.
- **ZippyFish**: in questo caso non è proposta una simulazione, ma una realizzazione pratica della funzione di inseguimento, attuata attraverso due motori collegabili alle porte di I/O, gestite direttamente dal programma (in particolare, porta con sé un'utilissima libreria, **servo.c**, che fornisce primitive ad alto livello per il comando dei motori). La sua funzione è di far inseguire alla telecamera, montata su un supporto pan&tilt servocontrollato, un pesce di gomma giallo – in pratica una versione con due gradi di libertà nello spazio, anziché su un piano, del progetto "Follow my socks" dell'LDRA! Con la differenza, sostanziale, che il sistema non comporta interfacciamento a Saphira, al robot Pioneer o ad un PC, ma è interamente gestito dalla scheda CognaChrome. Nonostante la semplificazione dovuta all'abbandono di behaviour per ostacoli e simili, può essere un utile punto di partenza per qualsiasi applicazione del sistema ARC.

4.4. Lo sviluppo di codice e la compilazione

Una volta installato e configurato il pacchetto ARC, abilitato l'uso della porta seriale, ecc., è possibile sviluppare codice per la scheda CognaChrome.

Il codice può essere scritto e compilato con target differenti: direttamente per la scheda (dotata di processore 68332), oppure per il PC (intel x86). In quest'ultimo caso la libreria del Kernel linkata simula l'ambiente hardware CognaChrome per il debug su PC senza la necessità di un emulatore 332.

La compilazione avviene tramite il tool **arcc**, il quale si occupa tra le altre cose di selezionare il Kernel adatto al target indicatogli (creando le "define __332__" o "__386__"); arcc crea in pratica un Makefile che poi passa al compilatore **gcc** per generare il codice macchina (formato S-Record, .s19, oppure a.out).

Il programma si può quindi caricare nella RAM della scheda tramite il comando "ramload" dell'ARC Serial Interaction Program, oppure direttamente nella EEPROM ("romload"), affiancato all'ARC Kernel ed eventualmente alle Vision Libraries. Se il codice non è già compilato, ramload e romload accettano anche il sorgente C: semplicemente invocano arcc prima dell'upload.

I dettagli sono descritti, nuovamente, nell' "ARC Development System User's Guide".

5. Conclusioni e sviluppi futuri

Fondamentalmente resta ancora da soddisfare l'obiettivo primario di questa esercitazione: valutare le effettive capacità (e, di conseguenza, anche la desiderabilità dell'acquisto) del sistema di sviluppo completo ARC.

Intanto se ne può ipotizzare l'uso:

- Le funzioni ad alto livello fornite sono orientate al tracking di forme, quindi principale sbocco applicativo del sistema sarà lo sviluppo ed implementazione di altre routine, diverse e più complesse dell'identificazione dei BLOB attualmente svolta dalle routine standard di CognaChrome, nonché l'estensione delle stesse: compensazione degli effetti della luce ambientale, dell'angolo di visuale, ecc.; riconoscimento di forme colorate complesse; pattern recognition; costruzione e mantenimento di mappe; ecc. Ovviamente l'applicazione sarà limitata dalla penalizzante capacità di calcolo e di memoria del calcolatore on-board (MCU 68332@16MHz privo di FPU, 256Kb/1Mb RAM, 256Kb EPROM), ma la flessibilità è garantita dalla possibilità di compilare sorgenti C nel codice oggetto per tale processore tramite GCC sotto Unix, eventualmente testabile già sul PC, e soprattutto (si spera!) attraverso il debugger fornito dal "Full" ARC Development System.
- Le porte di I/O libere disponibili sulla scheda microcontroller di CognaChrome la rendono sfruttabile per il controllo diretto di sensori e motori di un robot; la costruzione di un telaio e poca elettronica di potenza sarebbero sufficienti all'allestimento di robot dotati di visione, probabilmente meno flessibili del Pioneer (che deve ringraziare più che altro il comodissimo Saphira) ma sicuramente più specializzati e – soprattutto! – ben più economici.

Bibliografia e riferimenti

[1] Newton Research Labs: “ARC Development System User’s Guide”, Edition 1.1, 1995-2001

[2] Newton Research Labs: “Cognachrome Vision System User's Guide”, Edition 2.0, 1996

[3] Konolige, K.G.: “Saphira, a robot architecture”, 1999

[4] ActivMedia Robotics: “Saphira operations and programming manual”, Version 6.2, 1999

[5] Motorola Semiconductors: “MC68332TS/D rev.2 – MC68332 32-Bit Modular Microcontroller Data Sheet”, 1993-1996

[6] Gatta et al.: "Follow my rabbit + Manuale Cognachrome", University of Brescia, 2000

[7] Newton Research Labs (*CognaChrome & ARC*):
`http://www.newtonlabs.com`
`http://www.newtonlabs.com/academic-home.html`

CognaChrome Vision System:
`http://www.newtonlabs.com/cog.htm`

CognaChrome user’s guide:
`http://www.newtonlabs.com/cognachrome/manual/cognachrome_toc.html`

ARC:
`http://www.newtonlabs.com/arc.htm`

Vesta SBC332 board:
`http://www.newtonlabs.com/arc/vesta.html`

[8] ActivMedia Robotics (*Pioneer*):
`http://robots.activmedia.com`

Fast-Track Vision System:
`http://robots.activmedia.com/fast-track/`

Saphira for Pioneer robots:
`http://robots.activmedia.com/Saphira/`

[9] SRI International (*Saphira*):
`http://www.ai.sri.com`

SAPHIRA - A Robot Architecture:
`http://www.ai.sri.com/~konolige/saphira/index.html`

Kurt Konolige's Home Page:
`http://www.ai.sri.com/~konolige/`

[10] Vesta Technology Embedded Controllers (*Vesta SBC332 board*)
`http://www.vestatech.com/`

MC 68332 programming resources:
`http://www.ee.uwa.edu.au/~braunl/eyebot/ftp/68332/`

[11] University of Maryland, Space Systems Laboratory (*SCAMP*)
`http://www.ssl.umd.edu/`

SCAMP – Supplemental Camera and Maneuvering Platform:
`http://www.ssl.umd.edu/homepage/Projects/SCAMP/Project_overview.html`

[12] Università degli Studi di Brescia, Facoltà di Ingegneria, Laboratorio Didattico di Robotica Avanzata (*Speedy* e *Tobor*):
`http://bsing.ing.unibs.it/~cassinis/`

Indice