

Video Elaboration & Transmission Library

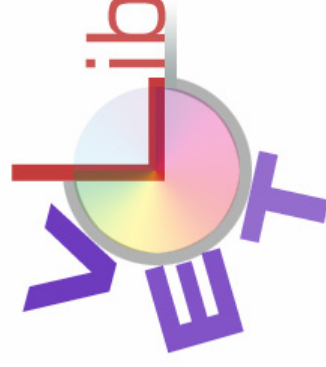
VETLib

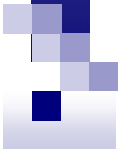
21/03/2006
PRESENTATION BUILT 1.0.2.83

Laureando: **Alessandro Polo**

Relatore: **Francesco De Natale**
Correlatore: **Nicola Conci**

A.A. 2004-2005





Analisi: Contesto

Elaborazione software di segnali video digitali:

- Sviluppo di filtri e moduli generici (developers);
- Sviluppo di software per l'utente finale (consumer);
- Dimostrazione didattiche (students);

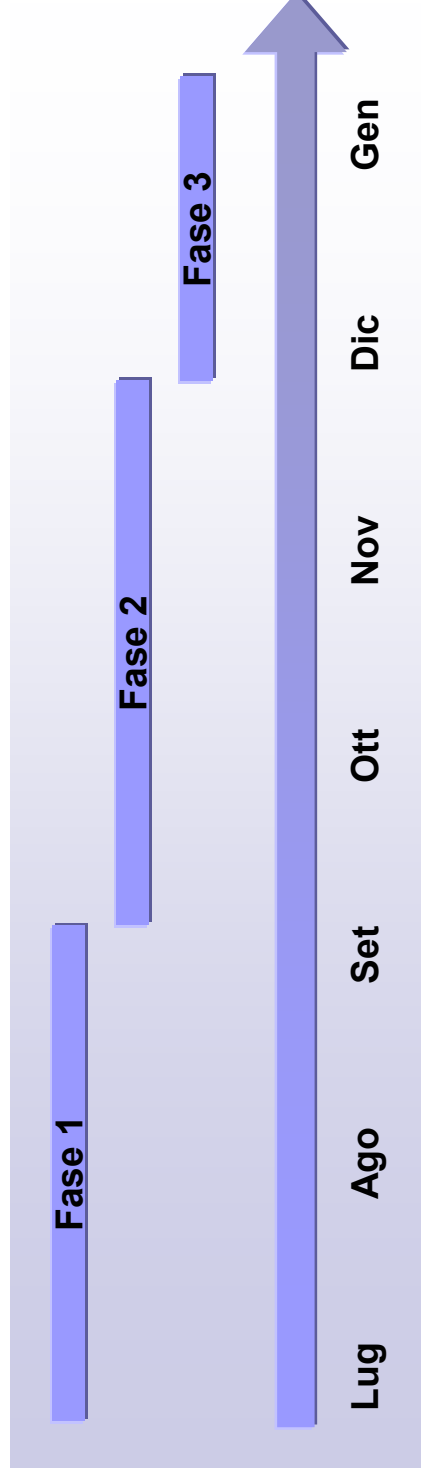
Stato dell'Arte:

- Molte librerie di image processing; (molte open source)
- Poche librerie dedicate al video processing o al Real Time Imaging; (la maggior parte proprietarie)
- Spesso le implementazioni sono ottimizzate per uno scopo specifico e difficilmente de-contestualizzabili.

 Libreria Ex Novo

Progetto: Linea di Lavoro

- Fase 1
 - Analisi stato dell'arte, specifiche di progetto, definizione obiettivi;
 - Progettazione di alto livello (identificazione interfacce principali);
 - Definizione e implementazione classi principali;
- Fase 2
 - Sviluppo di oggetti comuni, filtri ed esempi dei packages;
 - Interfaccia con librerie esterne (MPEG, XVID, Quicktime, DirectX, v4l, ..);
 - VETLib Workshop;
- Fase 3
 - Debugging & Testing, Aggiornamento documentazione;
 - VETLib Package Studio;



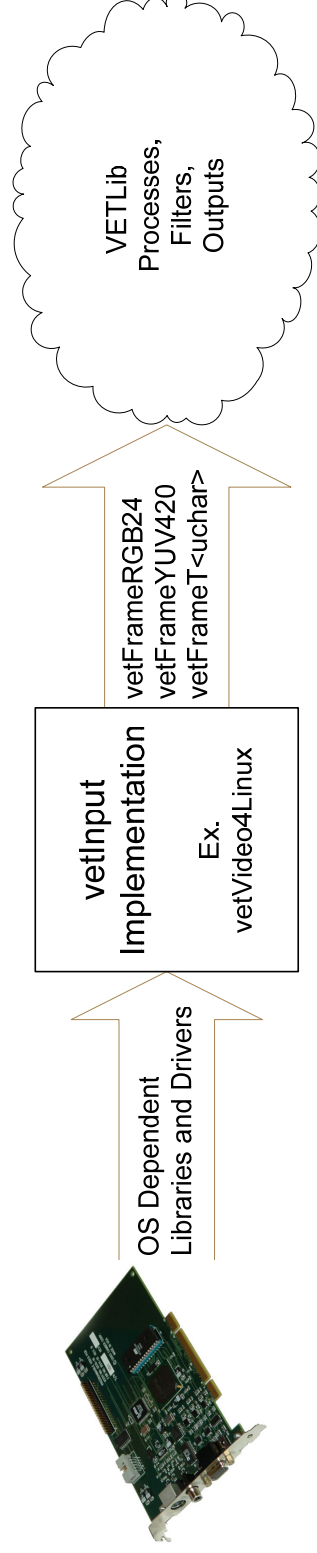


Analisi: Obiettivi

- Sviluppare una libreria modulare, scalabile, multi-piattaforma.
- Definire le interfacce necessarie per uniformare le relazioni tra i componenti (presenti e futuri).
- Sfruttare le peculiarità OOP del C++ per estendere la libreria e aggiornare moduli esistenti.
- Fornire allo sviluppatore un set di oggetti e metodi comuni, standard e compatibili con altri progetti.
- Interfacciare periferiche e codifiche specifiche con uno o più formati standard interni a VETLib in modo trasparente e scalabile.
- Integrare altre librerie esistenti (es. codifiche) purché siano distribuite con licenza open source.

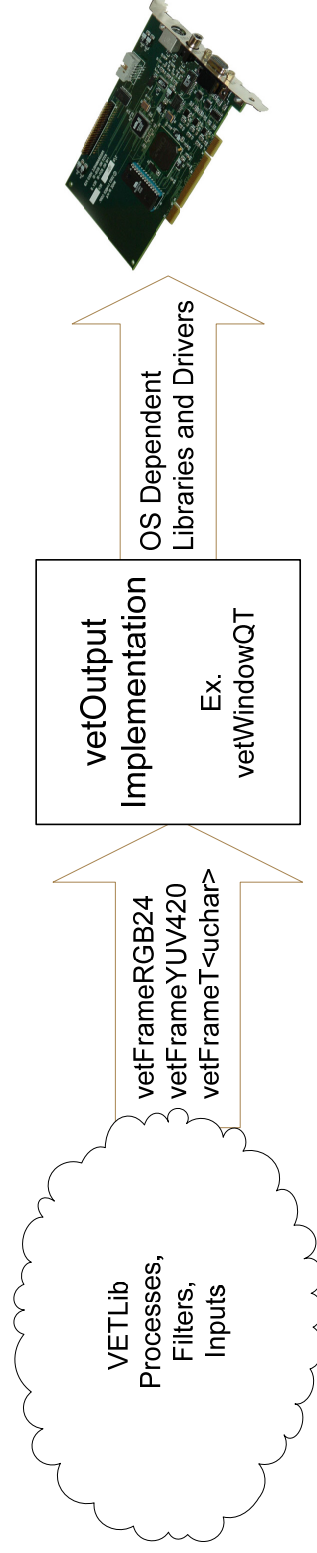


Flusso Dati esterno



Implementazione di vetInput

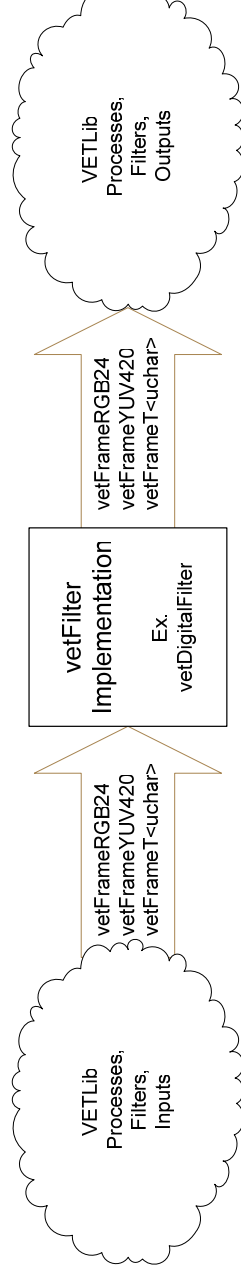
I dati vengono acquisiti da un driver o una libreria esterna, convertiti nello standard di VETLib e indirizzati in uscita quando richiesto.



Implementazione di vetOutput

I dati vengono letti in ingresso, convertiti nel formato richiesto e inviati all'interfaccia successiva come un driver o una libreria esterna.

Flusso Dati interno



vetFilter eredita vetInput e vetOutput incorporandone le capacità di interazione.

Implementazione di vetFilter

I dati acquisiti in ingresso vengono elaborati, salvati in un buffer e indirizzati in uscita quando richiesto. [Ex. vetDigitalFilter, vetFilterGeometric]



Moduli implementati

- Sistemi NIX
 - Acquisire da device compatibili con video4linux (usb cams, tv tuners, ..)
 - Decodificare: MPEG1-2, MPEG4 (XVID)
 - (de)Codificare: Quicktime (MOV)
 - (de)Codificare quasi tutti i formati immagine (+90)
 - Visualizzare immagini e video (QT, GTK)
- Sistemi Windows
 - Acquisire da device compatibili con DirectX (anche IEEE1394 OHCI)
 - Decodificare stream MPEG4 (XVID)
 - (de)Codificare quasi tutti i formati immagine (+90)
 - Visualizzare immagini (GDI, librerie esterne: QT, GTK)
 - VETLib WorkShop (VETLib front-end)



Documentazione

- Manuale *[italiano]*
 - Presentazione Libreria;
 - Utilizzo ed Estensione;
 - WorkShop e plugins;
- Leggimi (“Readmes”) *[inglese]*
 - Testuali, nella radice delle cartelle;
 - Struttura classica dei software complessi;
- Sito Web *[inglese]*
 - Informazioni dei “Readmes” in HTML;
 - downloads, accesso online alla documentazione e alla distribuzione;
- Inclusa nel sorgente *[inglese]*
 - Classi, metodi, variabili commentate;
 - Generazione automatizzata di reference (DoxyGen);

Esempio: MPEG Player



```
[..]  
mpegSource.load("VETLib-Logo.mpg");
```

Caricamento video

```
vetWindowGTK myWin(mpegSource.getWidth(), mpegSource.getHeight());  
myWin.show();
```

Inizializzazione finestra

```
double offset = 0;  
long sleeptime = (long) (1000 / mpegSource.getVideoFrameRate()) - 10;  
while (i++ < 100)
```

Ritardo tra frames (ms)

```
{  
    offset = vetUtility::getTime_usec();  
    mpegSource >> img24;  
    myWin << img24;
```

Redirezione frames

```
    vetUtility::vetSleep( sleeptime - (long)(vetUtility::getTime_usec()-offset)/1000 );
```

```
}  
[..]
```

Sincronizzazione frame rate

Tempo di elaborazione (us)

file: ./tests/app_vetLinuxMPEGPlayer.cpp

Esempio: Live Player (+ Flipping Filter)

```
[..]
vetVideo4Linux cap ( "/dev/video0" );
vetWindowGTK myWin(cap.getWidth(), cap.getHeight() );
vetFrameRGB24 imgRGB24 ( cap.getWidth(), cap.getHeight() );
vetFilterGeometric filter;
filter.setParameters().setRunMode(vetFilterGeometricParameters::FLIP_VERTICAL);

while ( !cap.EoF() && i++ < iMaxLoop)
{
    cap >> img24;           // cap.extractTo(img24);
    filter << img24;         // filter.importFrom(img24);
    filter >> img24;
    mywin << img24;         // mywin.importFrom( *filter.dump_buffer_RGB() );
}
[..]
```

Connessione periferica video

Controllo "End of File"

Redirezione frames

Redirezione frames ottima



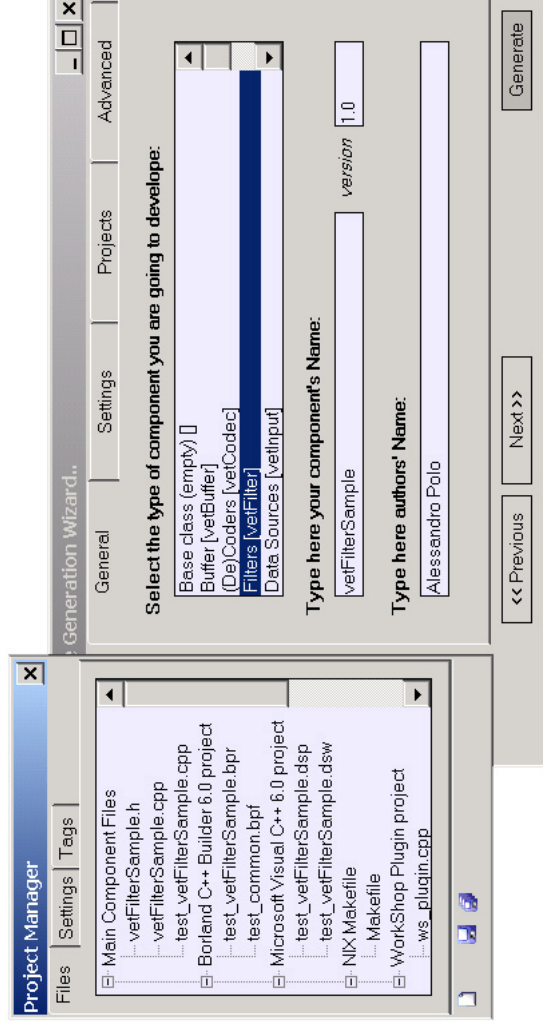
Pro & Contro

- Esperienze positive
 - Le specifiche previste sono state rispettate.
 - Semplicità sintattica, chiarezza del codice e utilizzo dei metodi ereditati dalla super-class, ottima astrazione dal punto di vista dell'applicazione;
 - Velocità e semplicità dello sviluppo di filtri e componenti, flessibilità per soluzioni specifiche;
 - L'utilizzo degli strumenti inclusi nella libreria permette RAD, VETLib Workshop permette di testare catene di filtri in modo veloce e pratico;
- Esperienze negative
 - La compilazione della libreria e lo sviluppo di moduli multi-piattaforma richiedono una certa esperienza di programmazione su Windows e specialmente su sistemi NIX;
 - La gestione della libreria è complessa, soprattutto il rilascio di una nuova release e la distribuzione.

VETLib Package Studio

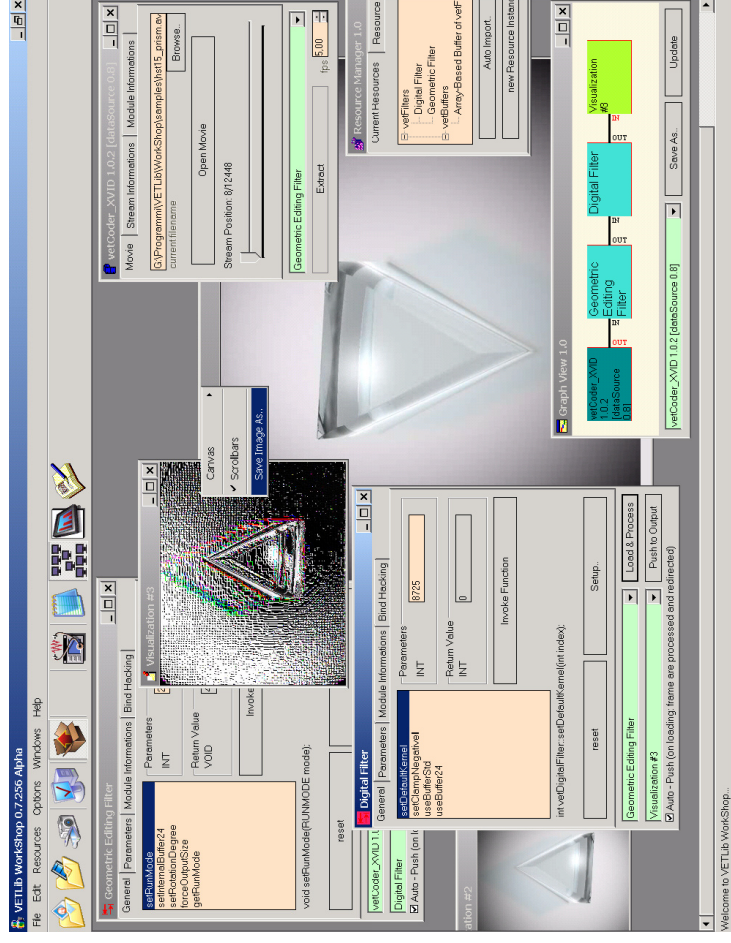
È uno strumento per sviluppatori di componenti VETLib.

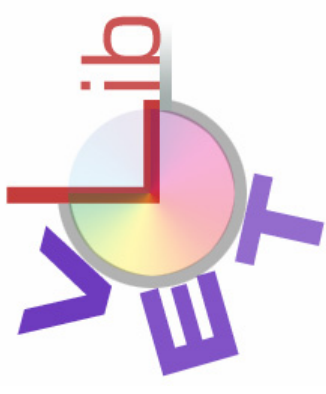
- Si basa su moduli predefiniti e file di configurazione XML, l'aggiornamento e l'aggiunta dei moduli supportati non richiedono la ricompilazione;
- Il Wizard semplifica la configurazione iniziale del progetto;
- Sono generati i files sorgente e tutti i files dei progetti selezionati.



VETLib WorkShop

- VETLib WorkShop è un'applicazione sviluppata in C++ .NET
- WorkShop è destinato all'utilizzo pratico e al test dei componenti VETLib.
- WorkShop è in grado di caricare in modo dinamico i componenti (DLL).
- Il formato di streaming interno è vetFrameRGB24.





More informations @

<http://lnx.ewgate.net/vetlib/>

Contact alessandro.polo@ewgate.net

vetlib@ewgate.net

more projects at <http://www.ewgate.net/research/>