

hrv_lan – interface gráfica para aquisição de vídeo e frequência cardíaca em uma LAN

1. Introdução/Justificativa

O objetivo deste software é capturar vídeo de webcams de computadores conectados a uma LAN. Paralelamente à captura, o sinal de ECG do usuário também deve ser capturado.

As principais características do software são:

- 1 - Possui janela do operador, onde se controla o início da captura.
- 2 - Permite estabelecer a rotina de captura. tempo em NVNM etc.
- 3 - Possui janelas para os subjects.
- 4 - Deve ser capaz de exibir vídeos pre-gravados, vídeos de câmeras transmitidos via streaming e uma tela vazia com alguma cor.
- 5 - Deve ser capaz de salvar o sinal de ECG.
- 6 - Deve ser capaz consultar as cameras disponiveis no localhost e nos demais hosts da rede. Deve ser possível localizar automaticamente os servidores de câmeras na rede com uma determinada porta aberta.

2. Classes

2.1. Classe LanDevice

A classe LanDevice é uma camada que expõe as funcionalidades dos dispositivos de captura disponíveis na LAN. Os dispositivos em que estamos interessados são as câmeras e o sensor cardíaco Polar H10. Possui os seguintes métodos:

`start_server`

Inicia o servidor que disponibiliza os dispositivos do localhost para a rede.

`stop_server`

Interrompe o servidor.

`list_servers`

Envia um sinal de broadcast para a rede para localizar todos os servidores ativos.

`list_devs_local(dev_type)`

Lista os dispositivos do localhost, com valor de dev_type em {CAM, POLAR}.

`list_devs_host(dev_type, ip_addr)`

Lista os dispositivos do host de IP especificado, com valor de dev_type em {CAM, POLAR}.

`list_devs_lan(dev_type)`

Lista os dispositivos da LAN, com valor de dev_type em {CAM, POLAR}.

`handle = stream_start(dev_type, dev_id, ip_addr)`

Inicia streaming de vídeo ou de ECG do dispositivo especificada. Retorna um handle.

`stream_show(dev_type, handle, window)`

Exibe streaming de vídeo ou de ECG, com handle especificado, em window.

`stream_save(dev_type, handle, filename)`

Salva streaming de vídeo ou de ECG com handle especificado para arquivo.

2.2. Classe Routine

A classe Routine lê os arquivos que controlam a rotina de captura e exibição de um experimento. Esses arquivos estão em formato CSV, com valores separados por ponto e vírgula. A primeira coluna contém o tempo em segundos que uma ação é executada, a segunda coluna contém a ação e as demais contém os parâmetros ou operandos.

Abaixo está um exemplo de arquivo de rotina de captura.

```
0.0; message; all; "Welcome!"
0.0; clear; all; #FFFF00
5.0; message; all; "Please watch this video"
5.0; play; all; "videos/instructions.mp4"
30.0; message; all; "No View No Motion"
45.0; message; all; "No View Motion"
60.0; message; s1; "Please gesticulate"
60.0; message; s2; "Spontaneous Imitation"
60.0; show; s1; c2
60.0; show; s2; c1
90.0; message; s2; "Induced Imitation"
120.0; message; all; "Please stop. Thank you!"
120.0; clear; all; #FFFF00
...
```

Abaixo está a lista de comandos suportados. Cada comando recebe dois operandos.

Instrução	Op. 1	Op. 2	Comentário
message	User	String	Show message on screen
show	User	Cam	Show camera content on video canvas
clear	User	Color	Clear screen and fill with color
play	User	String	Play video and show on video canvas

Os Valores de que User e Cam podem assumir são

```
User = {s1, s2, all}
Cam = {c1, c2}
```

onde s1 corresponde à janela do subject 1, s2 corresponde à janela do subject 2, all

corresponde a todas as janelas dos subjects, c_1 corresponde à câmera do subject 1 e c_2 corresponde à câmera do subject 2.

2.3. Classe WinOp

A classe WinOp cria a janela usada pelo operador do sistema. Expõe as funcionalidades dos dispositivos, câmeras e sensores cardíacos, da LAN ao usuário, permite definir a rotina de captura, selecionar os dispositivos a serem usados e iniciar a rotina de captura. Para evitar atrasos de comunicação, é definido um intervalo de tempo t em segundos. O comando de iniciar a rotina agenda o início da rotina nos clientes em hora atual mais t segundos. Idealmente, todas as máquinas envolvidas devem ter seus relógios sincronizados.

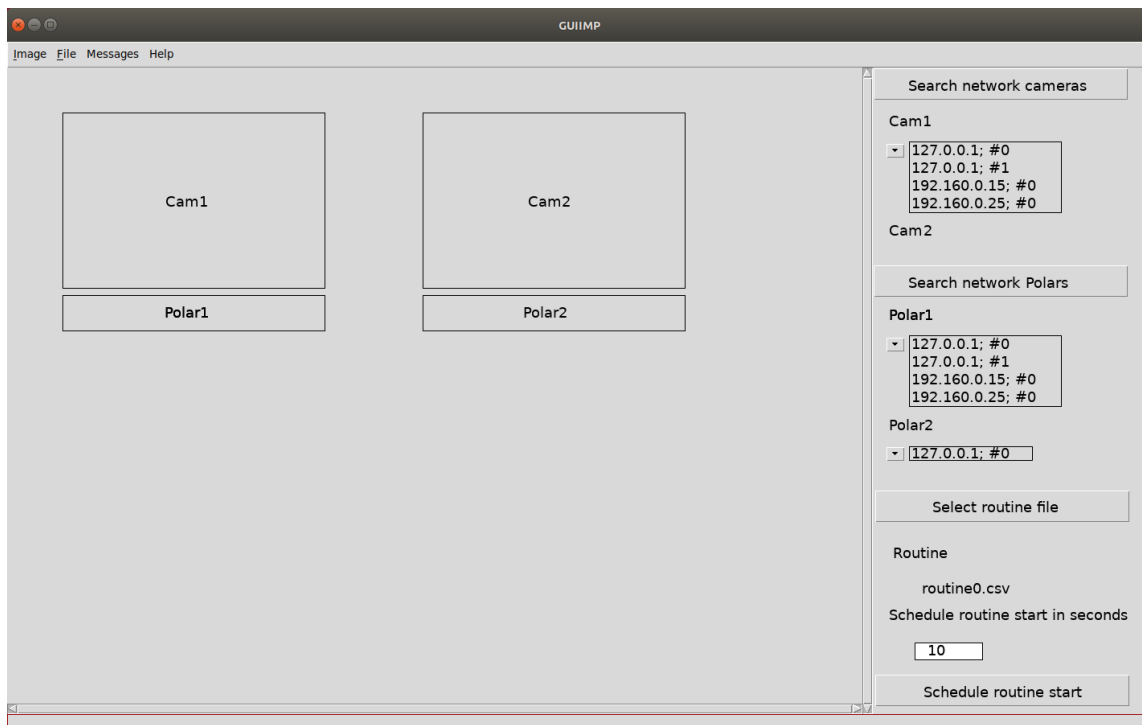


Figura 1 – Janela WinOp

A janela possui duas áreas principais. À direita está o canvas onde o conteúdo das câmeras selecionadas é exibido. À esquerda está o menu com as funcionalidades disponíveis.

O primeiro botão, *Search network cameras*, localiza os servidores rodando na LAN e povoa os menus para seleção de câmeras. Os menus *dropdown* permitem a seleção das câmeras, que começam a enviar o streaming assim que são selecionadas. O segundo botão faz o mesmo mas com os sensores cardíacos.

O terceiro botão, *Select routine file*, abre uma caixa de diálogo para que o operador selecione o arquivo contendo a rotina de captura.

Abaixo há uma caixa de texto para que o operador selecione um intervalo de tempo t em segundos. Ao clicar no botão *Schedule routine start*, é enviada a rotina de captura e o horário atual mais t segundos que a rotina deve ser iniciada. A janela deve possuir ao menos o menu abaixo.

- *Devices*

- *List cameras in localhost*
- *List cameras in LAN*
- *List cameras at IP*
- *List Polar H10 in localhost*
- *List Polar H10 in LAN*
- *List Polar H10 at IP*

2.4. Classe WinSubj

A classe WinSubj cria a janela usada pelo subject participante do experimento. É controlada por comandos enviados por WinOp via socket, mesmo que ambos os processos estejam no mesmo host. Exibe o conteúdo de alguma câmera da rede, mostra vídeos e mostra mensagens com instruções.

A janela deve possuir ao menos o menu abaixo.

- *Devices*

- *List cameras in localhost*
- *List Polar H10 in localhost*



Figura 2 – Janela WinSubj

3. Arquitetura

A Figura 2 mostra as camadas do sistema. A Figura 3 mostra como as janelas são distribuídas entre os hosts e como se comunicam pela rede.

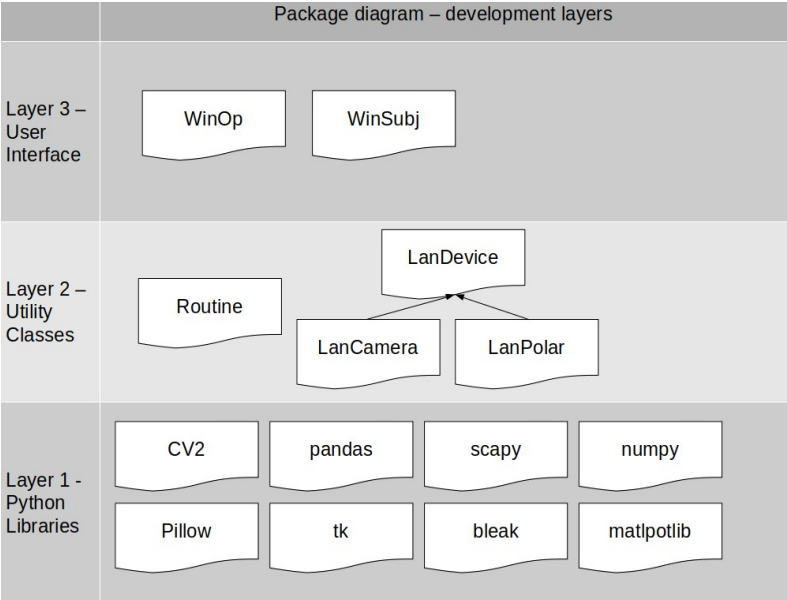


Figura 3 - Package diagram: deployment layers

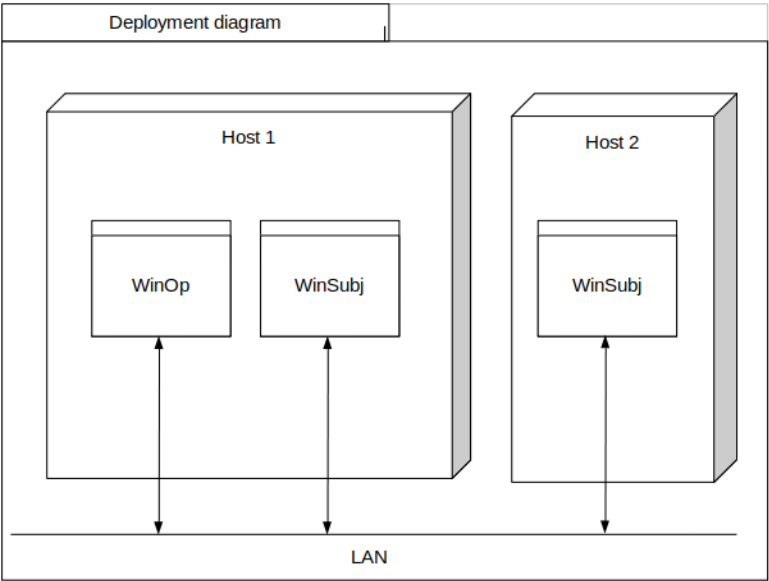


Figura 4 – Deployment diagram

4. Protocolo de comunicação

A janela WinOp envia comandos e consultas às janelas WinSubj rodando na LAN. Os comandos são

`query_servers`

Sinal de broadcast para encontrar os servidores ativos.

`query_server_cams`

Consulta as câmeras disponíveis de um certo servidor

`query_server_polars`

Consulta os sensores cardíacos disponíveis de um certo servidor

`select_cam`

Envia para servidor o id da câmera selecionada e atribui um rótulo em {c1, c2}. Inicia o streaming imediatamente para o canvas Cam1 ou Cam2.

`select_polar`

Envia para servidor o id do sensor cardíaco selecionada e atribui um rótulo em {p1, p2}. Inicia o streaming imediatamente para o canvas Polar1 ou Polar2.

`schedule_routine`

Envia para servidor o arquivo de rotina a ser executado e agenda o início da captura para algum horário nos próximos segundos.