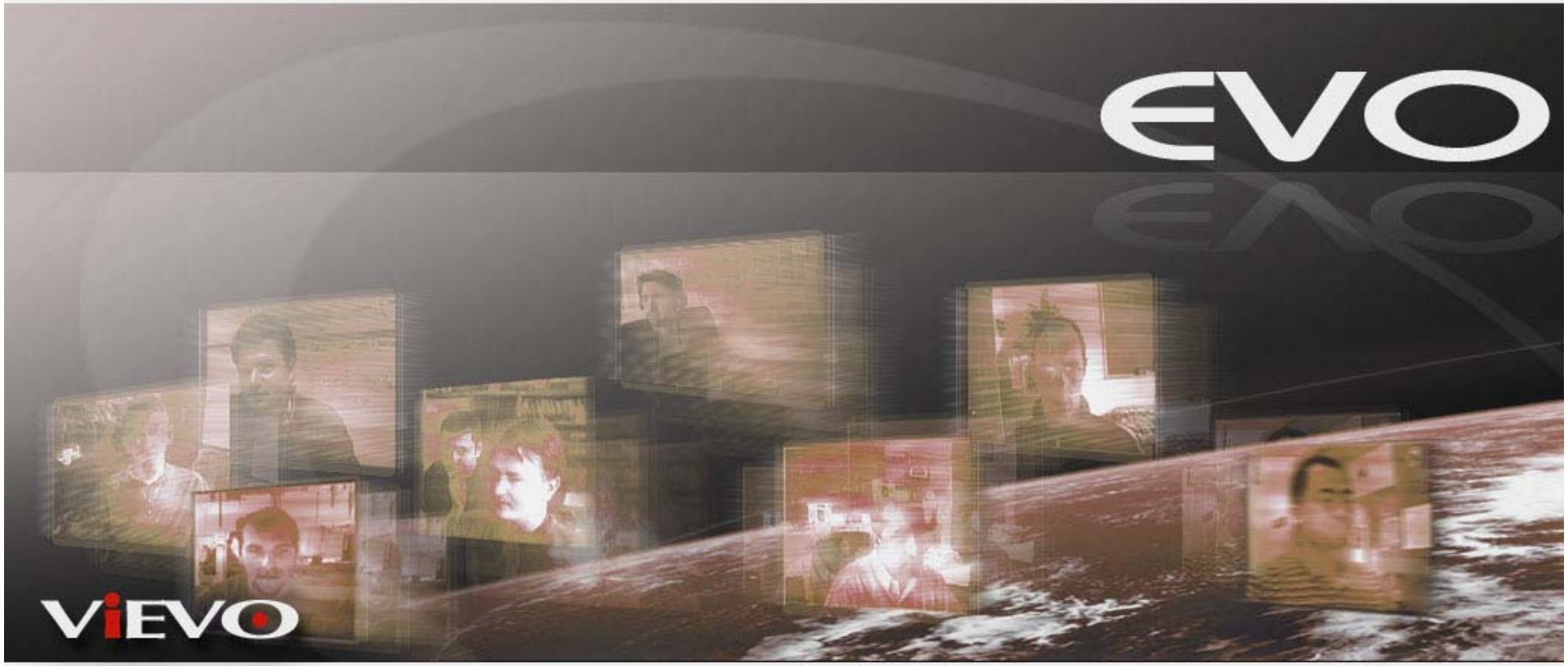
Using OpenGL and 3D to Manage Large Numbers of Video Conferencing Streams



Pavel FARKAS, Ph.D.
for
California Institute of Technology

From VIC (VRVS) to ViEVO (EVO)

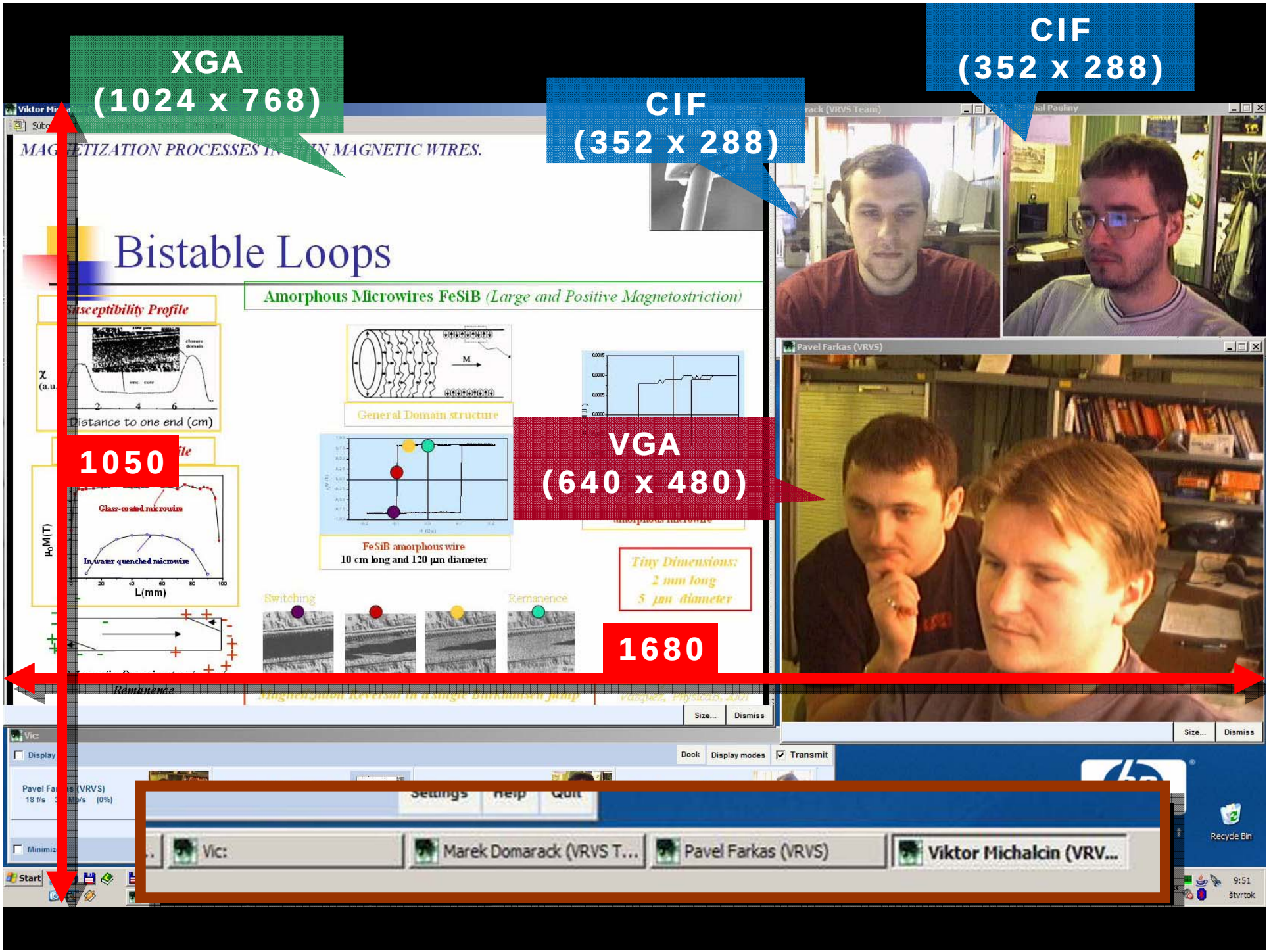
ith developing of video application
o develop a new video application
iEVO

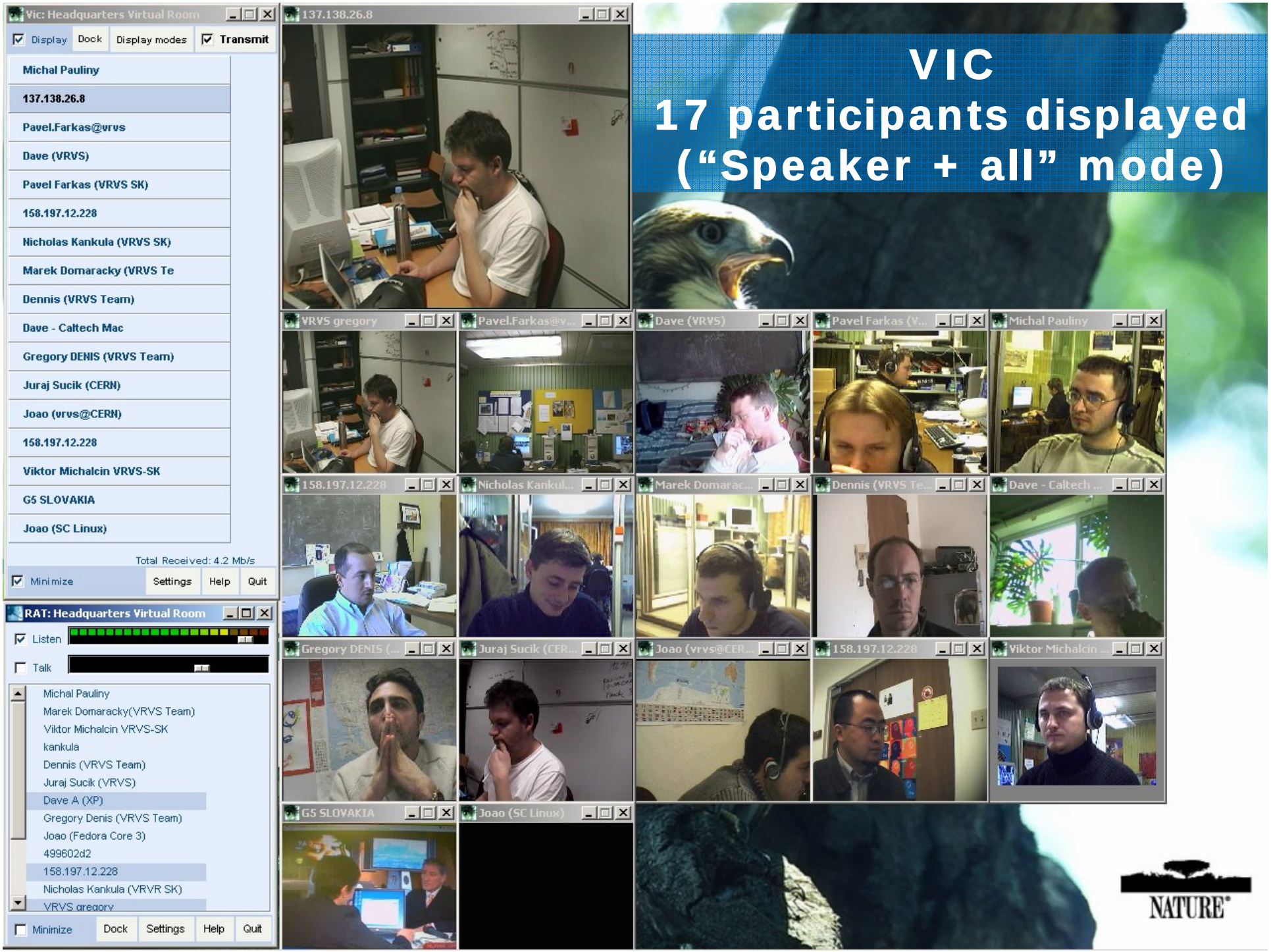
Display facilities of VIC (VRVS)

- each video displayed in separate window, what leads to following negative consequences:
  - crowded taskbar in case of larger number of video streams
  - management of display windows on a computer screen is uncomfortable – has been partially solved with automatic display modes

- strong linkage between physical video resolution and display size of video source
  - video source can be displayed only in its original size, or in two re-sampled versions: ¼ of original resolution (every second sample is taken) and 4x of original resolution (samples are duplicated in each direction) - main reason is to save maximum CPU resources for video encoding and decoding
  - computer screen is covered very soon by video windows in case of larger number of video streams

- limited possibilities to create visually attractive display functionality


XGA
(1024 x 768)
CIF
(352 x 288)
CIF
(352 x 288)
Bistable Loops
Amorphous Microwires FeSiB (Large and Positive Magnetostriction)
Susceptibility Profile
Distance to one end (cm)
General Domain structure
Glass-coated microwire
In-water quenched microwire
L(mm)
FeSiB amorphous wire
10 cm long and 120 μm diameter
Tiny Dimensions:
2 mm long
5 μm diameter
Switching
Remanence
Remanence
1050
VGA
(640 x 480)
1680
Pavel Farkas (VRVS)
Size...
Dismiss
Dock
Display modes
Transmit
Vic:
Marek Domarack (VRVS T...
Pavel Farkas (VRVS)
Viktor Michalcin (VRV...
Recycle Bin
Start
9:51
štvrtok


Vic: Headquarters Virtual Room
Display
Dock
Display modes
Transmit
Michal Pauliny
137.138.26.8
Pavel.Farkas@vrvs
Dave (VRVS)
Pavel Farkas (VRVS SK)
158.197.12.228
Nicholas Kankula (VRVS SK)
Marek Domaracky (VRVS Te
Dennis (VRVS Team)
Dave - Caltech Mac
Gregory DENIS (VRVS Team)
Juraj Sucik (CERN)
Joao (vrvs@CERN)
158.197.12.228
Viktor Michalcin VRVS-SK
G5 SLOVAKIA
Joao (SC Linux)
Total Received: 4.2 Mb/s
Minimize
Settings
Help
Quit
RAT: Headquarters Virtual Room
Listen
Talk
Michal Pauliny
Marek Domaracky(VRVS Team)
Viktor Michalcin VRVS-SK
kankula
Dennis (VRVS Team)
Juraj Sucik (VRVS)
Dave A (XP)
Gregory Denis (VRVS Team)
Joao (Fedora Core 3)
499602d2
158.197.12.228
Nicholas Kankula (VRVR SK)
VRVS gregory
Minimize
Dock
Settings
Help
Quit
137.138.26.8
VRVS gregory
Pavel.Farkas@v...
Dave (VRVS)
Pavel Farkas (V...
Michal Pauliny
158.197.12.228
Nicholas Kankul...
Marek Domarac...
Dennis (VRVS Te...
Dave - Caltech ...
Gregory DENIS (...
Juraj Sucik (CER...
Joao (vrvs@CER...
158.197.12.228
Viktor Michalcin ...
G5 SLOVAKIA
Joao (SC Linux)
VIC
17 participants displayed
("Speaker + all" mode)
NATURE

ViEVO (EVO) and OpenGL solution

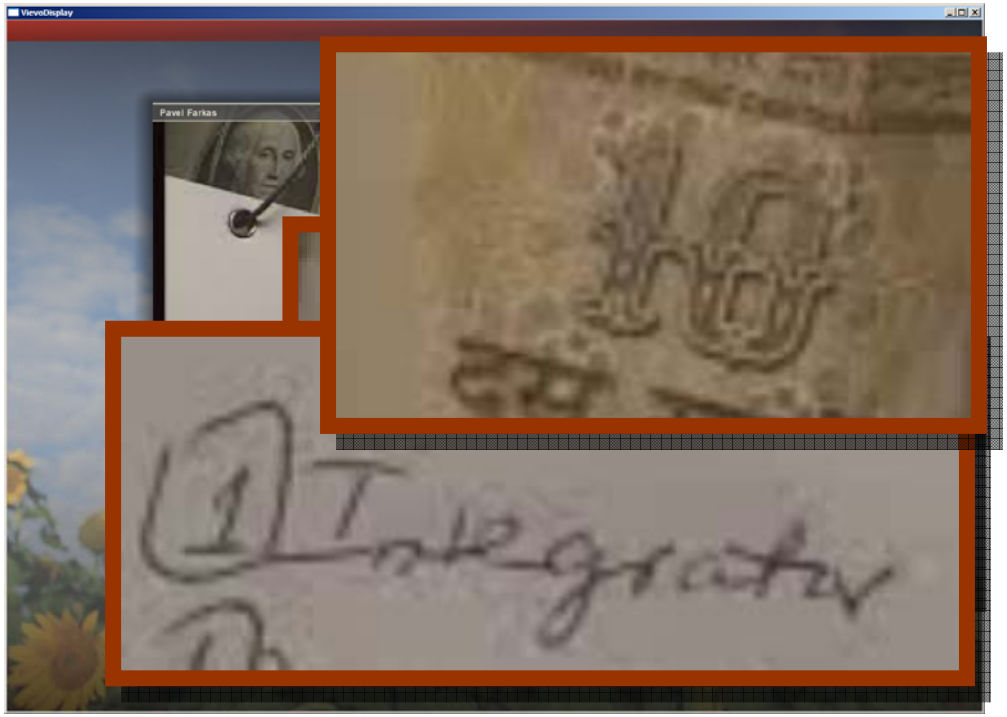
- easy and straightforward solution is to display all video sources in one common display window managed by OpenGL
  - only one display window on the taskbar
  - user can place the window on arbitrary position on the screen with arbitrary window size – content will be managed automatically by OpenGL in accord with selected display mode

- video resolution and display size are not linked anymore
  - texture is created from each decoded video frame and then it is mapped to rectangle of selected (arbitrary) size. Textures are stretched or shrinked by OpenGL as necessary and this is applied using the current texture filter (performed by graphic hardware, CPU is not used)
  - space on the screen can be used effectively

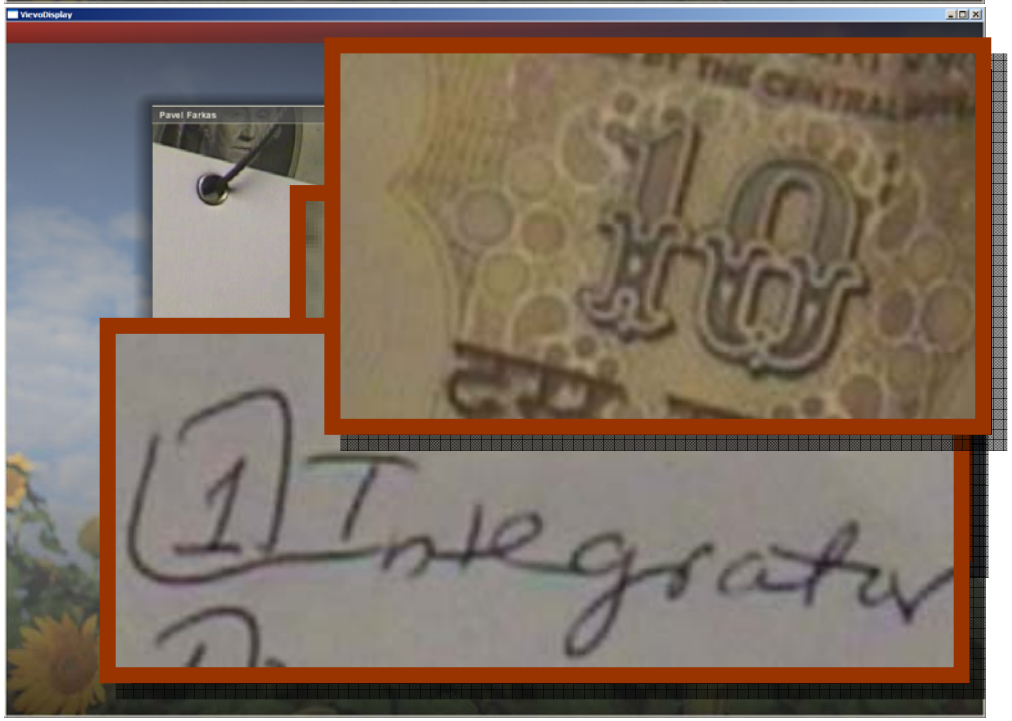
- OpenGL allows us to create any display mode in 3D which is hardware accelerated – many possibilities to create visually very attractive content

Comparison No. 1

Video with CIF resolution (new OpenGL solution)



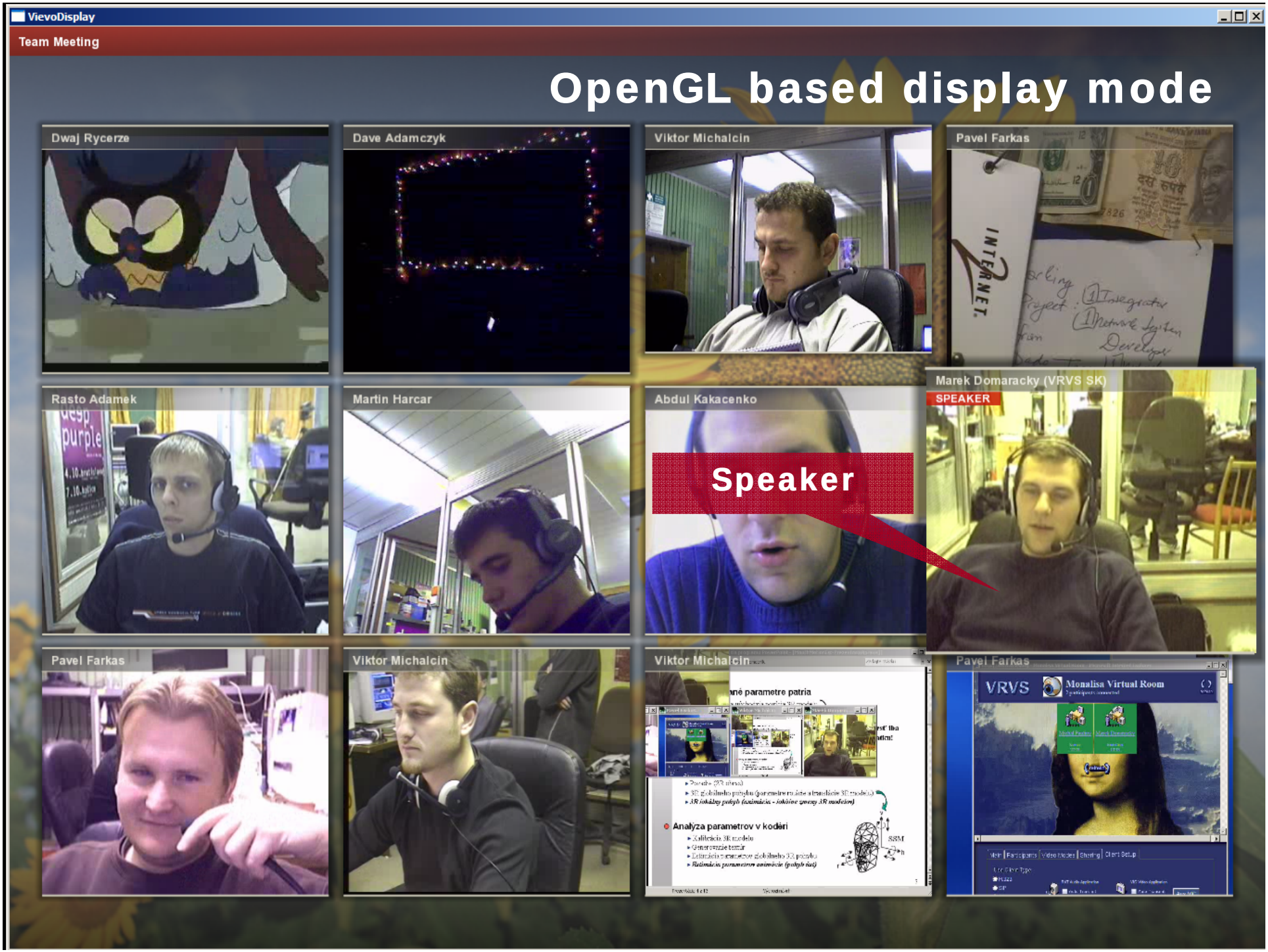
Video with VGA resolution (new OpenGL solution)



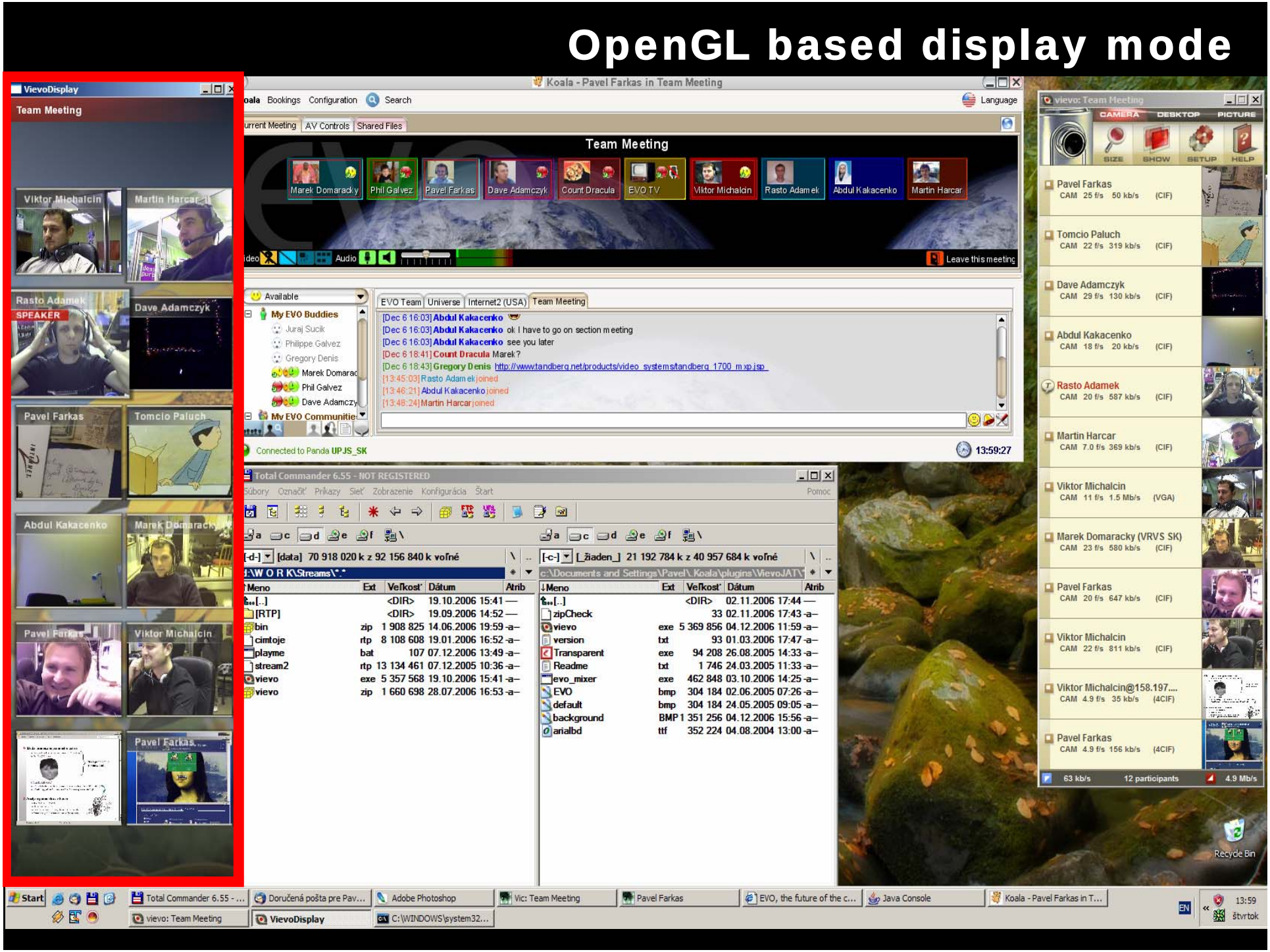
Comparison No. 2

Video with CIF resolution enlarged to 4CIF
(old VIC solution)

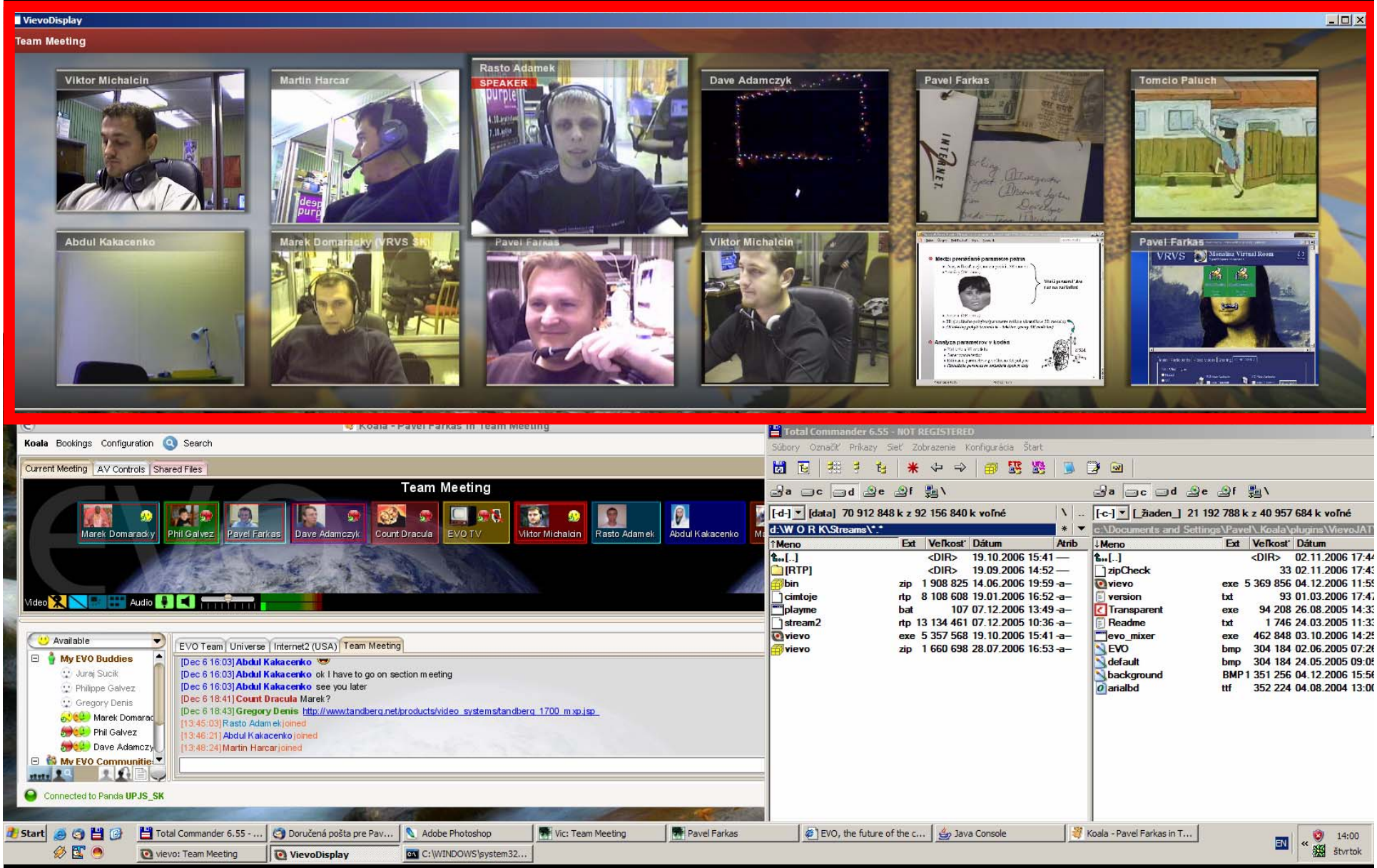
Video with CIF resolution enlarged to 4CIF
(new OpenGL solution)


VievoDisplay
Team Meeting
OpenGL based display mode
Dwaj Rycerze
Dave Adamczyk
Viktor Michalcin
Pavel Farkas
Rasto Adamek
Martin Harcar
Abdul Kakacenko
Speaker
Marek Domaracky (VRVS SK)
SPEAKER
Pavel Farkas
Viktor Michalcin
Viktor Michalcin
Pavel Farkas
VRVS
Monalisa Virtual Room
Analýza parametrov v kodéri

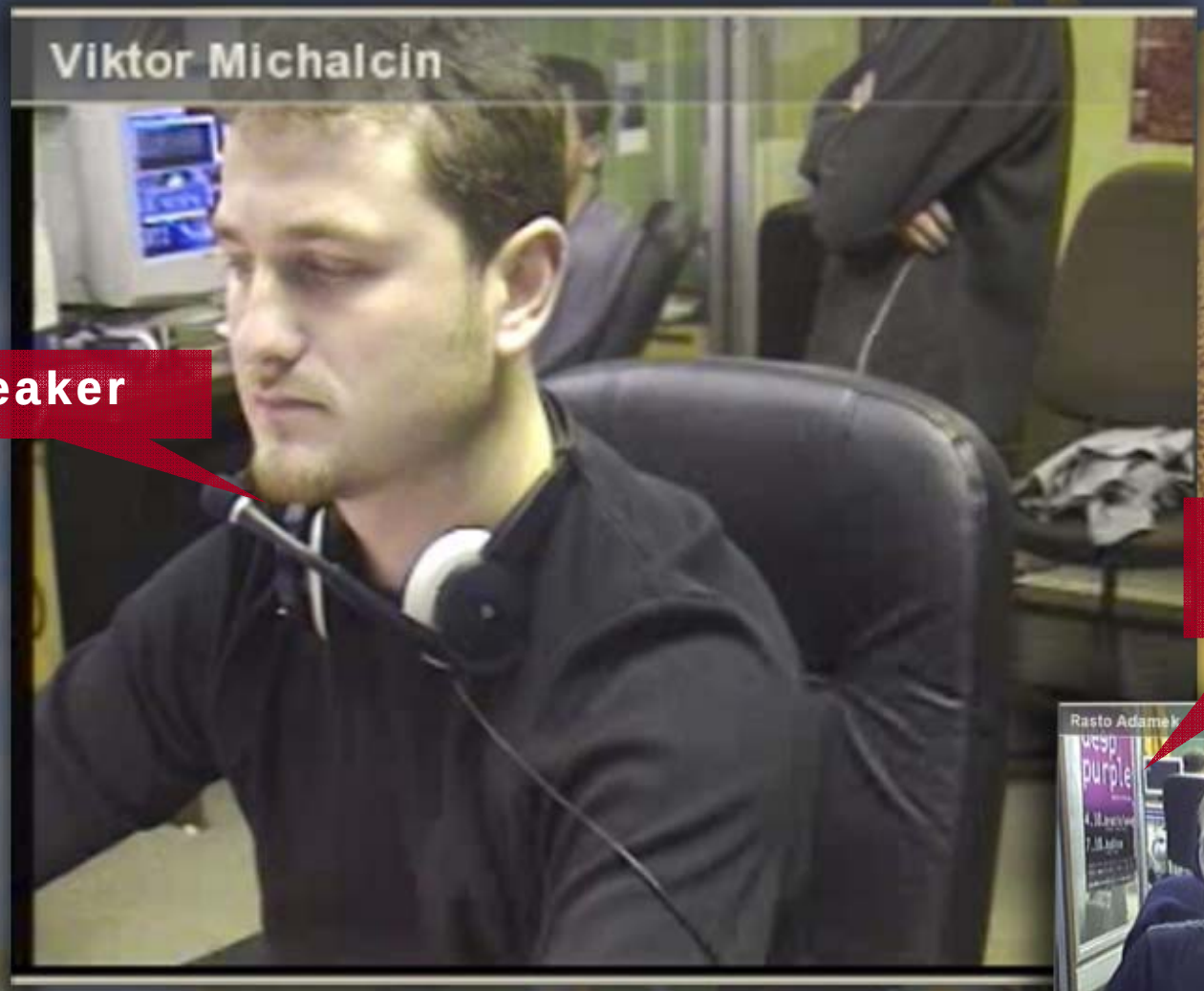
OpenGL based display mode



OpenGL based display mode



Ongoing and future work



Speaker

Local user's video

"Speaker" display mode

Ongoing and future work

"Debate" display mode

Ongoing and future work

Presentation

Presenter

Microsoft PowerPoint - [Prezentácia programu PowerPoint - [MouthMotion5ec-Prezentacia1prava]]

Súbor Úpravy Prehľadávať Okno Pomocník Zadajte otázku

- **Medzi prenášané parametre patria**
  - Tvar, veľkosť a východzia pozícia 3R modelu
  - Textúry (2R obraz)

Stačí preniesť iba raz na začiatku!

  - Pozadie (2R obraz)
  - 3R globálneho pohybu (parametre rotácie a translácie 3R modelu)
  - ***3R lokálny pohyb (animácia - lokálne zmeny 3R modelov)***

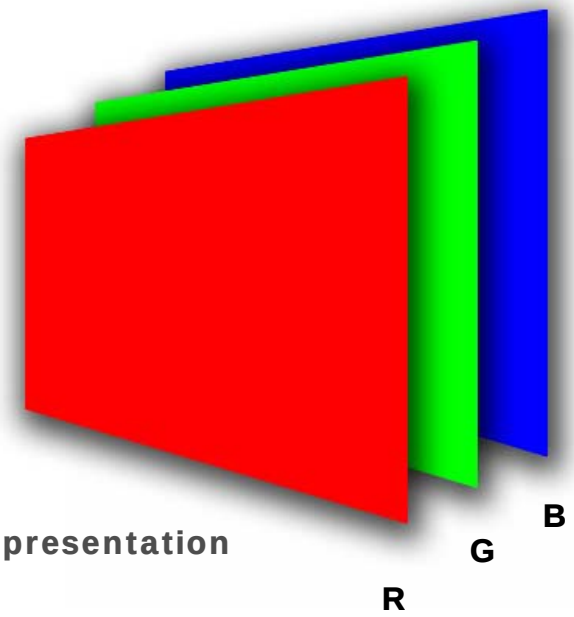
- **Analýza parametrov v kodéri**
  - Kalibrácia 3R modelu
  - Generovanie textúr
  - Estimácia parametrov globálneho 3R pohybu
  - ***Estimácia parametrov animácie (pohyb úst)***

Prezentácia 4 z 13 Výchozí návrh

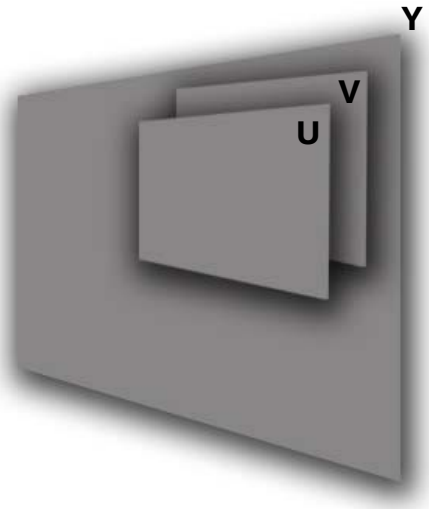
“Presentation” display mode

RGB and YUV color spaces

- the human visual system is **less sensitive to color** than to luminance (brightness)
- in RGB color space **all three colors are equally important** and all stored at the same resolution



RGB representation



YUV (4:2:0) representation – **requires half as many bits as the RGB representation**

- more efficient way is to separate luminance from color information and **represent luminance with higher resolution than color information** – YUV (YCrCb) color model

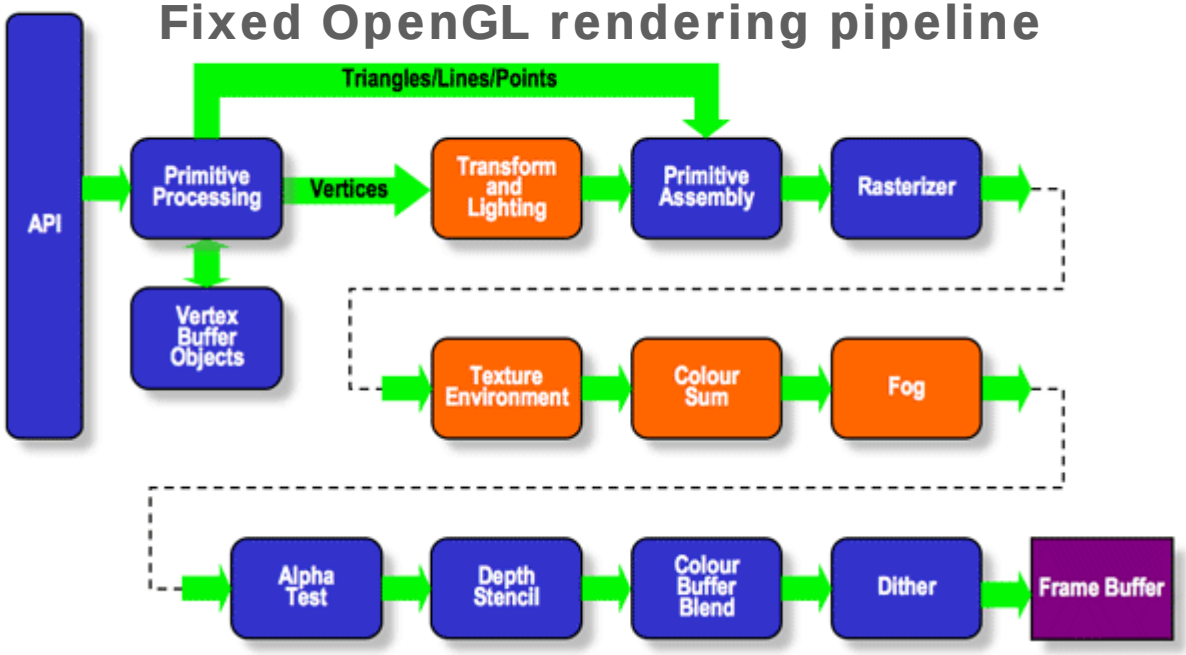
RGB versus YUV textures

- video displaying application requires continuous updating of video textures (unlike, for example, games, where textures can be preloaded at the beginning of the level)
- copying of large amount of data from PC memory to the graphic hardware memory can have negative impact on application performance

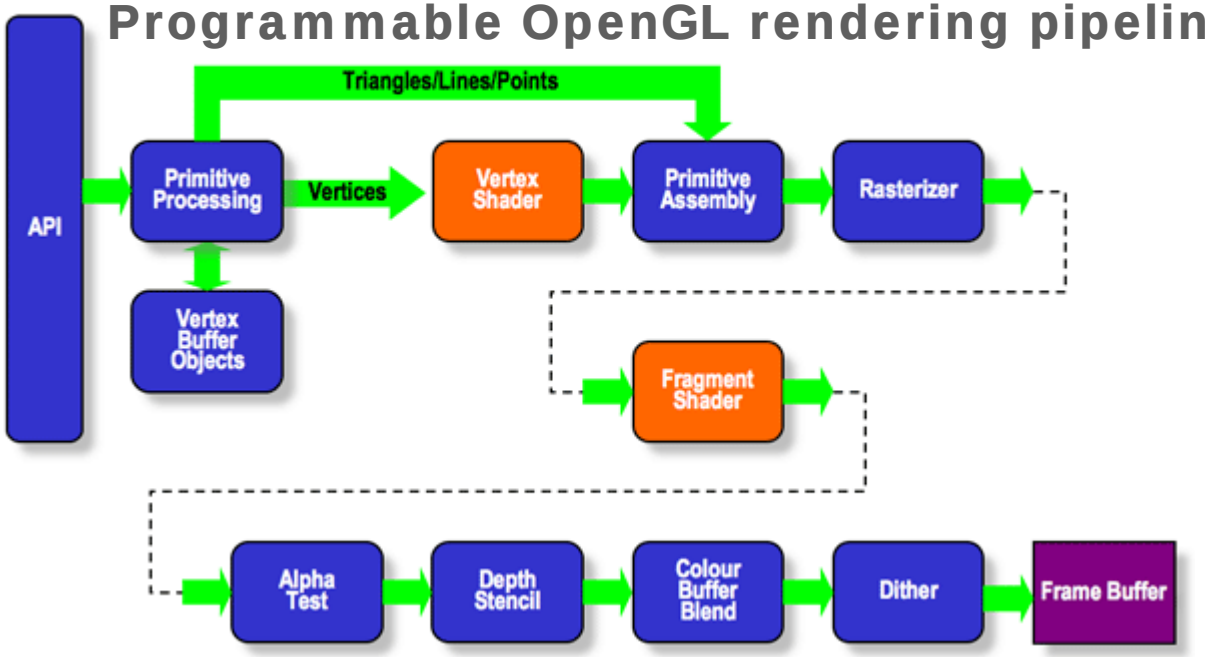
Can we speed up this process or make it more effective?

- solution are YUV textures (not natively supported by OpenGL) + utilization of programmable OpenGL pipeline (vertex and fragment shaders)
- benefits:
  - only half amount of data is transferred from PC memory to graphic card memory compare to RGB textures
  - saves the graphic card memory, 3x less memory usage compare to RBG textures
  - more effective memory utilization
  - YUV -> RGB conversion is performed by GPU – saves CPU!!

Fixed OpenGL rendering pipeline

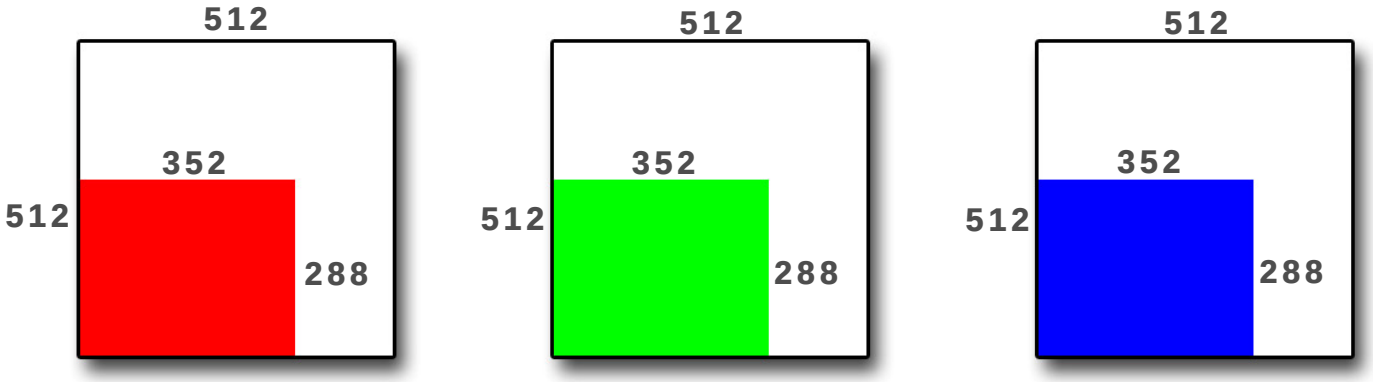


Programmable OpenGL rendering pipeline



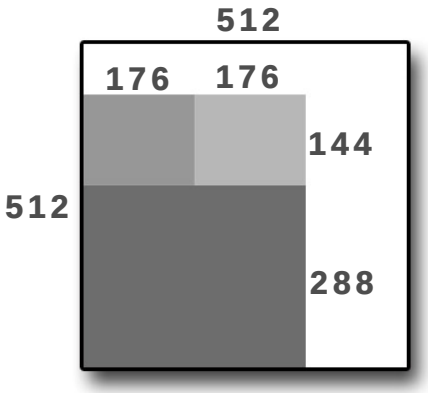
RGB versus YUV textures

glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, 512, 512,
0, GL_RGB, GL_UNSIGNED_BYTE, textureData);



RGB texture

glTexImage2D(GL_TEXTURE_2D, 0, GL_LUMINANCE, 512, 512,
0, GL_LUMINANCE, GL_UNSIGNED_BYTE, textureData);



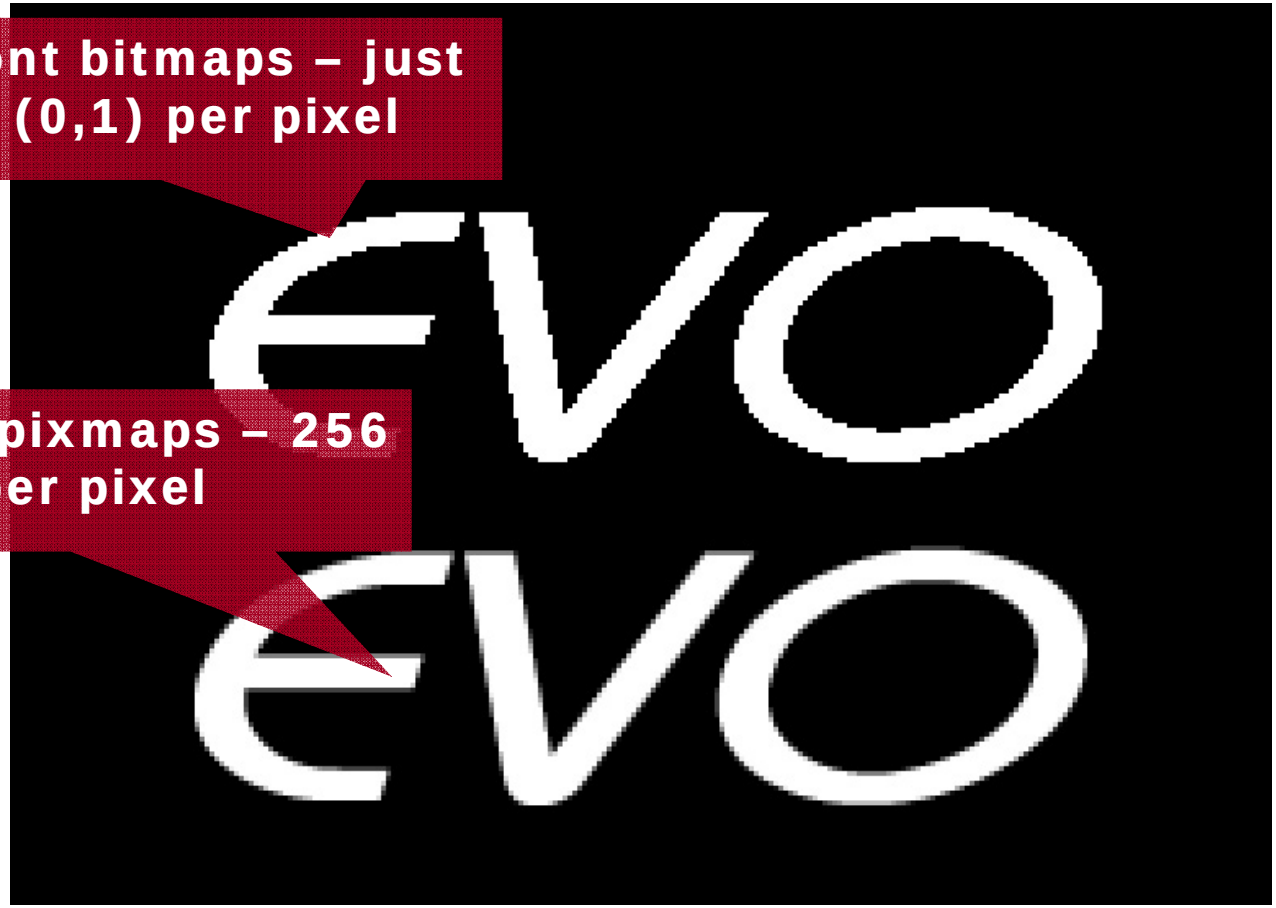
YUV texture

Font engine for OpenGL

• **FreeType library (Win32, Mac OS, Linux) - a software font engine that is designed to be small, efficient, highly customizable, and portable while capable of producing high-quality output - www.freetype.org**

WGL 2D font bitmaps – just 2 levels (0,1) per pixel

Freetype font pixmaps – 256 levels per pixel



Conclusions

- **interactivity** with OpenGL content – buttons, controls
- allows us to **remove old** Tcl/Tk based **GUI**
- next step - **porting the solution to MAC and Linux** platforms
- implement **advanced 3D models** of real meetings/conferences (audience in auditorium, participants sitting around a table, etc)
- all is valid also for MS Direct3D technology, OpenGL has been chosen for its multiplatform support

Thank you for your attention