Multi-perspective Images from Real World Scenes

Leonardo K. Sacht
leo-ks@visgraf.impa.br

Visgraf - IMPA

June 30, 2009

Main Reference

M. Agrawala, D. Zorin, T. Munzner:

Artistic Multiprojection Rendering.

In Eurographics Rendering Workshop, 2000.

Artistic Uses of Multiple Projections

- Artistic expression;

Artistic Uses of Multiple Projections

- Artistic expression;
- Representation of the scene.

Artistic expression: viewing anomalies

Title: *The Mystery and Melancholy of a Street*.

Author: Giorgio de Chirico (1888-1978).

Date: 1914.

Medium: Oil on canvas.

Style: Surrealist.

Artistic expression: viewing anomalies

Title: *Still Life with Fruit Basket*.

Author: Paul Cezanne (1839-1906).

Date: 1880-1890.

Medium: Oil on canvas.

Style: French Post-Impressionist.

Reference's Result

Our Result

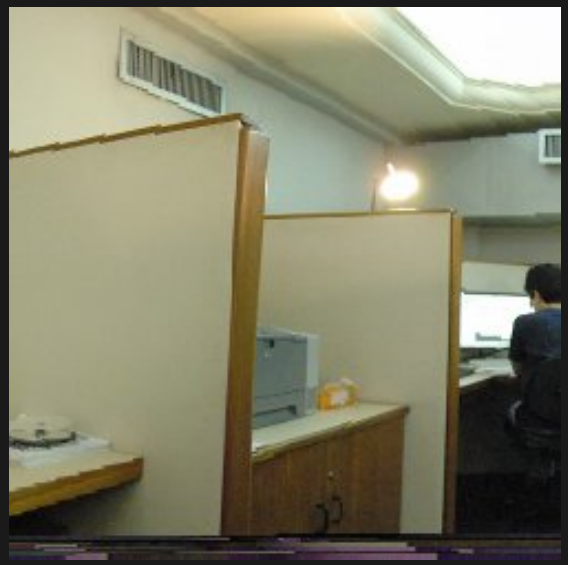
Representation: best views

Title: *Sir John Hawkwood*.

Author: Paolo Uccello (1397-1475).

Date: 1436.

Representation: best views



Our Result

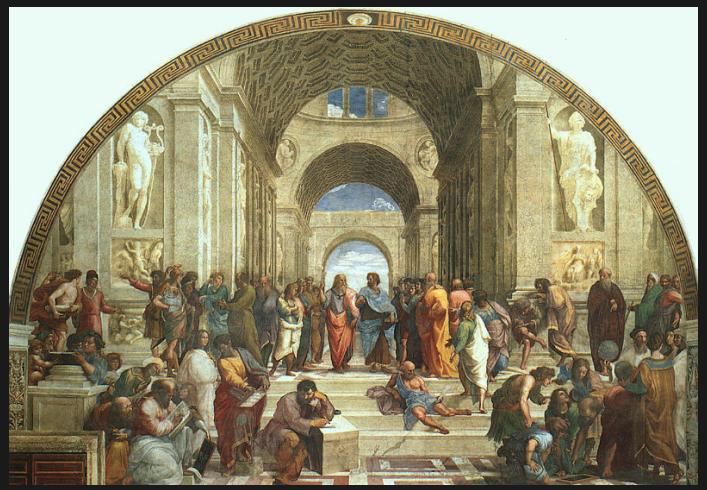
Representation: reducing wide-angle distortions

Title: *School of Athens*.

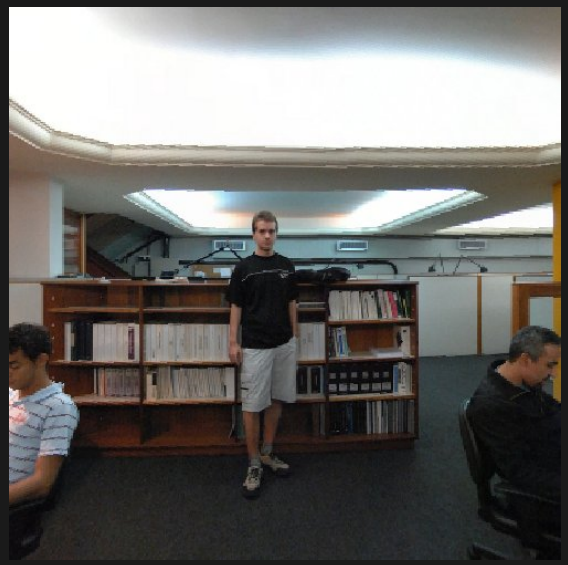
Author: Raphael Sanzio (1483-1520).

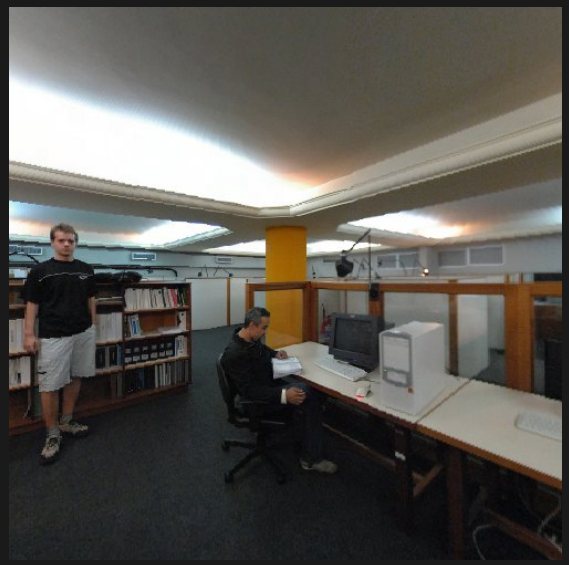
Date: 1510-1511.

Style: Intalian Renaissantist.



Representation: reducing wide-angle distortions





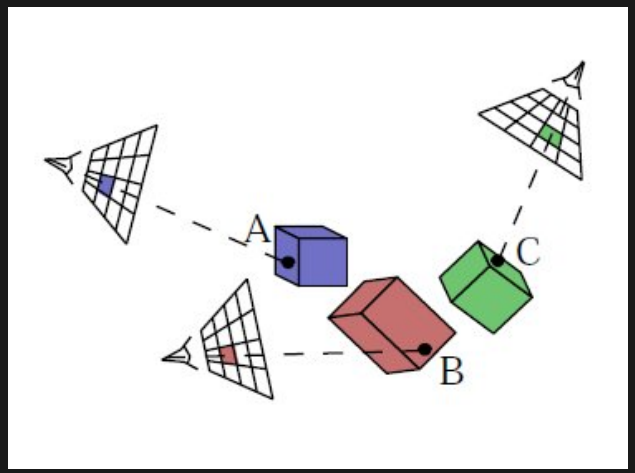
Our Result

Visibility Ordering

- For a single linear perspective, visibility is defined unambiguosly;

Visibility Ordering

- For a single linear perspective, visibility is defined unambiguosly;
- For multiperspective, however, the set of points that map to a same pixel on the image surface may be a union of straight lines.

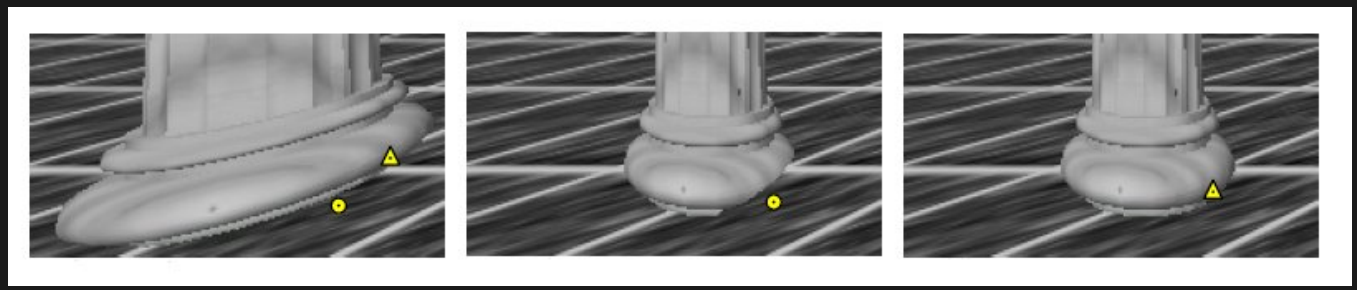


Master Camera

- For all points that would map to a same pixel on the multiperspective final image, we render the one that is closest to the center of projection of a chosen **Master Camera**. This process is called **depth compositing**;

Master Camera

- For all points that would map to a same pixel on the multiperspective final image, we render the one that is closest to the center of projection of a chosen **Master Camera**. This process is called **depth compositing**;
- But problems can occur:



Object-Level Occlusion Constraints

- The occlusions will be considered for each pair of objects on the scene;

Object-Level Occlusion Constraints

- The occlusions will be considered for each pair of objects on the scene;
- An occlusion constraint detection in relation to the Master Camera is performed;

Object-Level Occlusion Constraints

- The occlusions will be considered for each pair of objects on the scene;
- An occlusion constraint detection in relation to the Master Camera is performed;
- There are two possibilities for ambiguities: object A occludes object B in some regions and object B occludes A in other ones; objects A and B do not overlap on the master camera but overlap on the final result;

Object-Level Occlusion Constraints

- The occlusions will be considered for each pair of objects on the scene;
- An occlusion constraint detection in relation to the Master Camera is performed;
- There are two possibilities for ambiguities: object A occludes object B in some regions and object B occludes A in other ones; objects A and B do not overlap on the master camera but overlap on the final result;
- The user imposes hard constraints and decides abut the occlusion.

Compositing

- If two objects map to the same pixel and there is an occlusion contraint, we use it to solve visibiliy;

Compositing

- If two objects map to the same pixel and there is an occlusion contraint, we use it to solve visibiliy;
- If there are no occlusion constraints, use depth compositing;

Compositing

- If two objects map to the same pixel and there is an occlusion contraint, we use it to solve visibiliy;
- If there are no occlusion constraints, use depth compositing;
- If a cycle occurs (A occludes B, B occludes C and C occludes A), an arbitrary choice of what object will be seen is taken.

The Software

- I developed a software for the construction of multiperspective images;

The Software

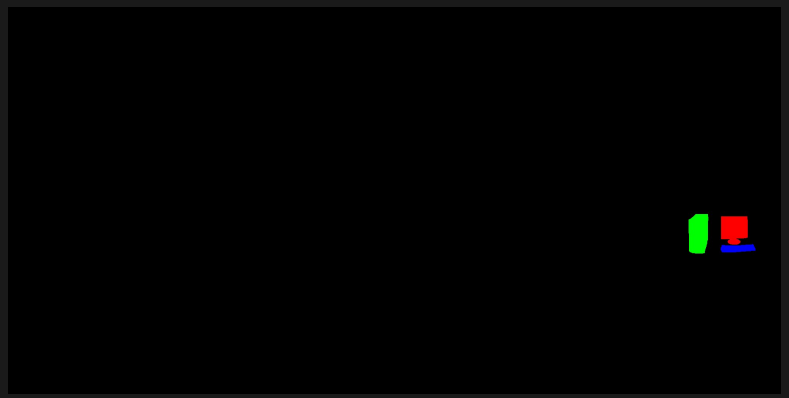
- I developed a software for the construction of multiperspective images;
- This sofwtare is based on windows and the interface was built with FLTK (Fast Light Toolkit). The programming languages used were C and C++.

The Software

- I developed a software for the construction of multiperspective images;
- This sofwtare is based on windows and the interface was built with FLTK (Fast Light Toolkit). The programming languages used were C and C++.
- It is divided in 4 windows and they're ran by a command line script.

Input Data

The user inputs a set of **equirectangular images** from a scene, taken from different points of view. For each image, he gives a corrsepondent image with marked objects and another wich is taken from a close point of view:



Input Data

Window 1 (Step 1)

The first window allows the user to choose

- the look-at vector of each camera;

Window 1 (Step 1)

The first window allows the user to choose

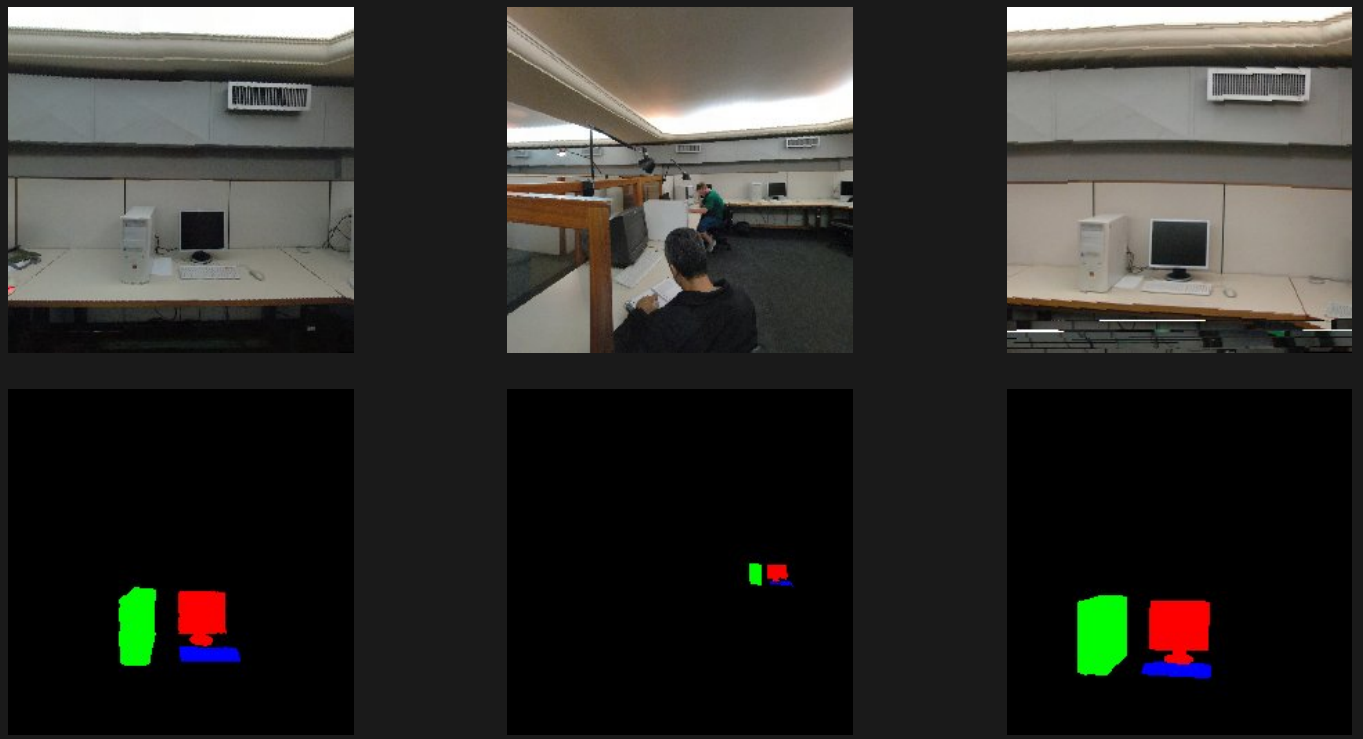
- the look-at vector of each camera;
- the zoom of each photograph;

Window 1 (Step 1)

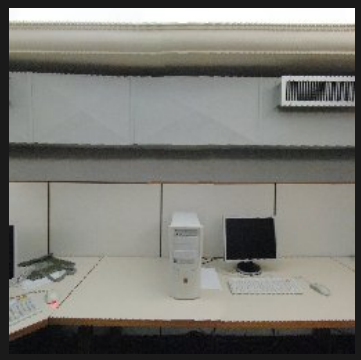
The first window allows the user to choose

- the look-at vector of each camera;
- the zoom of each photograph;
- and which camera will be the master camera;

Results produced by Window 1

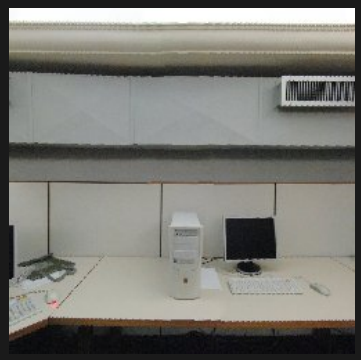


Intermediate Step - Computing stereo

- I decided to use an available algorithm to compute stereo (the pair of images is also output of window 1);

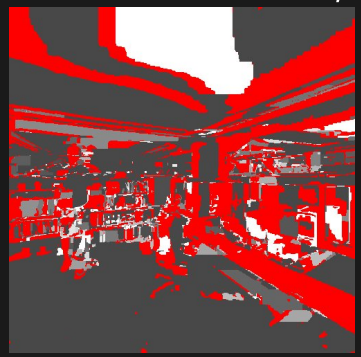
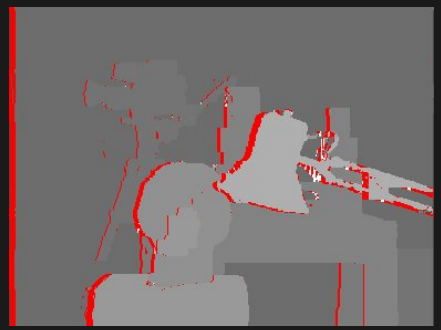
Intermediate Step - Computing stereo



- I decided to use an available algorithm to compute stereo (the pair of images is also output of window 1);
- The information returned by the algorithm would be used for determining depth to the Master Camera;

Intermediate Step - Computing stereo

In most cases, the results are not satisfactory, what lets most of occlucions to the user to determine;

Step 2 - Initial Occlusions detection

- For each pair of objects A and B, determine $z_{min}(A)$, $z_{max}(A)$, $z_{min}(B)$, $z_{max}(B)$;

Step 2 - Initial Occlusions detection

- For each pair of objects A and B, determine $z_{min}(A)$, $z_{max}(A)$, $z_{min}(B)$, $z_{max}(B)$;
- If $z_{max}(A) < z_{min}(B)$, set A *occludes* B;

Step 2 - Initial Occlusions detection

- For each pair of objects A and B, determine $z_{min}(A)$, $z_{max}(A)$, $z_{min}(B)$, $z_{max}(B)$;
- If $z_{max}(A) < z_{min}(B)$, set A *occludes* B;
- If $z_{max}(B) < z_{min}(S)$, set B *occludes* A;

Window 2 (Step 3) - User occlusion setting

- The user see the Master Camera image and the corresponding objects and chooses the occlusions clicking on buttons;

Window 2 (Step 3) - User occlusion setting

- The user see the Master Camera image and the corresponding objects and chooses the occlusions clicking on buttons;
- Cycles are forced to not happen (if A occludes B and the user sets C to occlude A, then the program set that C occludes A).

Window 2 (Step 3) - User occlusion setting

- The user see the Master Camera image and the corresponding objects and chooses the occlusions clicking on buttons;
- Cycles are forced to not happen (if A occludes B and the user sets C to occlude A, then the program set that C occludes A).
- The output of this window is a matrix that represent the occlusion:

$$\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix} \Rightarrow A \text{ occludes } B \text{ and } C \text{ and } B \text{ occludes } C.$$

Window 3 (Step 4) - Local Camera setting

- The objects given as input are highlited on the images and user chooses a camera for each one;

Window 3 (Step 4) - Local Camera setting

- The objects given as input are highlited on the images and user chooses a camera for each one;
- On the main reference, the local cameras for each object were given also as input;

Window 3 (Step 4) - Local Camera setting

- The objects given as input are highlited on the images and user chooses a camera for each one;
- On the main reference, the local cameras for each object were given also as input;
- The output of this window is a matrix:

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \Rightarrow A \text{ from cam } 2 \; B \text{ from } 2 \text{ and } C \text{ from } 3 \; .$$

Step 5 and Window 4 - Compositing

- For each (local) object, align its bounding box with its corresponding bounding box on the master camera image;

Step 5 and Window 4 - Compositing

- For each (local) object, align its bounding box with its corresponding bounding box on the master camera image;
- For each pixel, if there's no object (with new coordinates) on this pixel, put the color from the Master Camera if there was no object there or paint it with the object's indicator color;

Step 5 and Window 4 - Compositing

- For each (local) object, align its bounding box with its corresponding bounding box on the master camera image;
- For each pixel, if there's no object (with new coordinates) on this pixel, put the color from the Master Camera if there was no object there or paint it with the object's indicator color;
- If there is one object, put the color of this object;

Step 5 and Window 4 - Compositing

- For each (local) object, align its bounding box with its corresponding bounding box on the master camera image;
- For each pixel, if there's no object (with new coordinates) on this pixel, put the color from the Master Camera if there was no object there or paint it with the object's indicator color;
- If there is one object, put the color of this object;
- If there is more than one object, use the occlusion constraints and put the color of the object that occludes the others;

Step 5 and Window 4 - Compositing

- For each (local) object, align its bounding box with its corresponding bounding box on the master camera image;
- For each pixel, if there's no object (with new coordinates) on this pixel, put the color from the Master Camera if there was no object there or paint it with the object's indicator color;
- If there is one object, put the color of this object;
- If there is more than one object, use the occlusion constraints and put the color of the object that occludes the others;
- A window displays the final result;

Post Processing

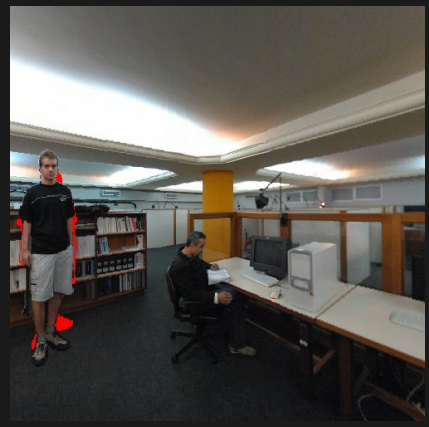
- The final image has holes caused by taking off objects from the Master Camera and putting nothing on its place.

Post Processing

- The final image has holes caused by taking off objects from the Master Camera and putting nothing on its place.
- Inpainting techniques will be usefull for post processing, but I did it manually.

Results






Panoramas

- Use of this program for panoramas: the wider the field of view the greater the need of correcting the perspective for the objects

Intermediate and post processing

- Find a more efficient way to compute order between objects automatically;

Intermediate and post processing

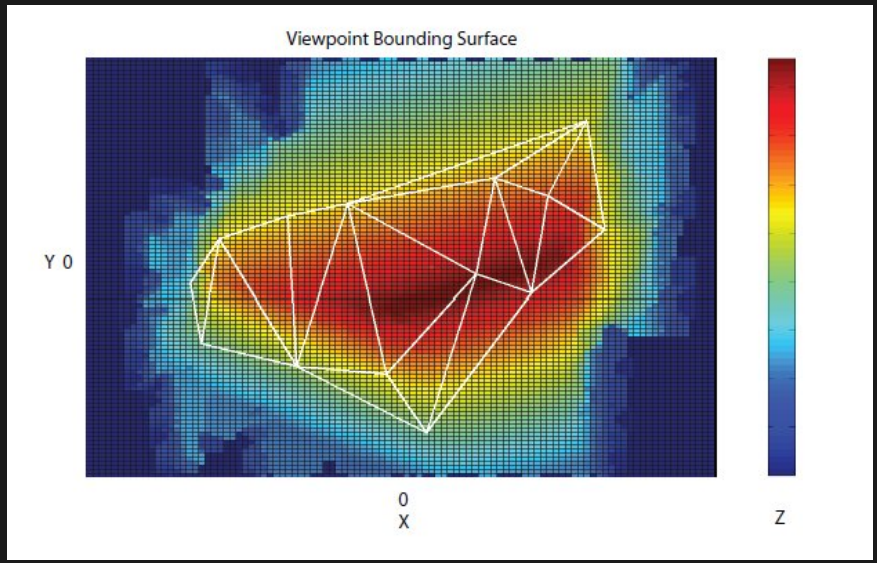
- Find a more efficient way to compute order between objects automatically;
- Inpainting technique for the result;

Intermediate and post processing

- Find a more efficient way to compute order between objects automatically;
- Inpainting technique for the result;
- Correction of disparities caused by putting different perspectives on the same image;

View Interpolation

Parallax Photography: Creating 3D Cinematic Effects from Stills



Thanks!

Check the website: *http* : //www.impa.br/ ∼ leo-ks/s3d