NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF RURAL AND SURVEYING ENGINEERS
DEPARTMENT OF TOPOGRAPHY
LABORATORY OF PHOTOGRAMMETRY

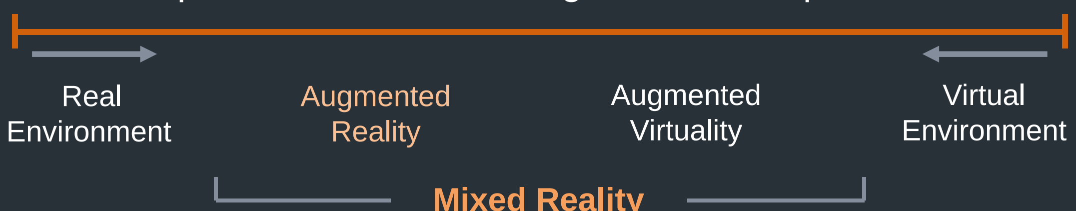
3D Visualization through Planar Pattern Based Augmented Reality

- Dr. **Charalabos Ioannidis**, Professor of Photogrammetry
- **Styliani Verykokou**, Surveying Engineer, PhD student

XXV INTERNATIONAL FIG CONGRESS
“ENGAGING THE CHALLENGES, ENHANCING THE RELEVANCE”
16-21 JUNE 2014, KUALA LUMPUR, MALAYSIA

AUGMENTED REALITY – AR (1/4)

- Enriches reality with computer generated information
- Enhances people’s perception of reality
- Adds mainly visual information in the real world, which has the dominant role
- Can potentially enhance all five senses
- Belongs to the technology of mixed reality (*Milgram et al., 1994*), according to which data that belongs to both the real and virtual world are presented as coexisting in the same place



AUGMENTED REALITY – AR (2/4)

AR Systems *(Azuma, 1997)*:

1. Combine real and virtual objects in a real environment
2. Allow real-time interaction
3. Register virtual objects in the three-dimensional space

- 1968: 1st AR system (*Sutherland*)
- 1992: the term "augmented reality" was coined (*Caudell & Mizell*)
- TODAY: applications in various fields

Medicine | Education | Navigation | Architecture – interior design

Commerce - advertising | Art – culture – tourism | The Army

Entertainment – sports | Task support | …

AUGMENTED REALITY – AR (3/4)

AR Applications in the field of surveyor

- Virtual reconstruction of half-ruined buildings, statues, or archaeological sites → real-time integration of their 3D models into the real world
- Visualization of constructions during their design phase
- Navigation (combination of AR and location-based services)

AUGMENTED REALITY – AR (4/4)

Augmented Reality

- Pattern based AR
- Outline AR
- Location-based AR
- Surface AR

Marker-based AR

Markerless AR

Tracking of the position and the orientation of the camera

Registration of the virtual object in the real world

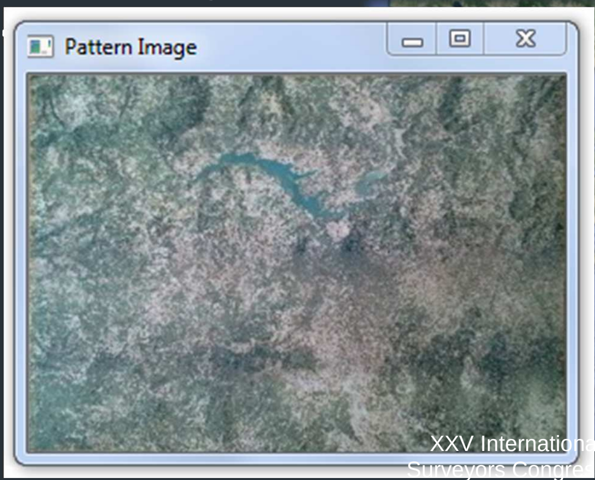
visual tracking

APPLICATION DEVELOPED

- **Application:** Planar pattern based markerless AR application
- **Purpose:** Visualization of the 3D anaglyph of a region through the screen of a PC, without the need of a 3D print-out of its DTM
- **Methodology:**
  - Initial data:
    - a pattern image
    - a 3D augmentation model in OBJ format
    - the interior orientation of the computer camera
  - Calculation of the exterior orientation of every video frame
  - Right rendering of the 3D model on a computer window whereby the background is the video frame

CASE STUDY

- **Pattern image:** a photograph of a printed orthoimage that depicts an area (18km x 12km) around the artificial lake of the river Ladonas in the Peloponnese, Greece
- **Augmentation model:** DTM of the region depicted in the orthoimage

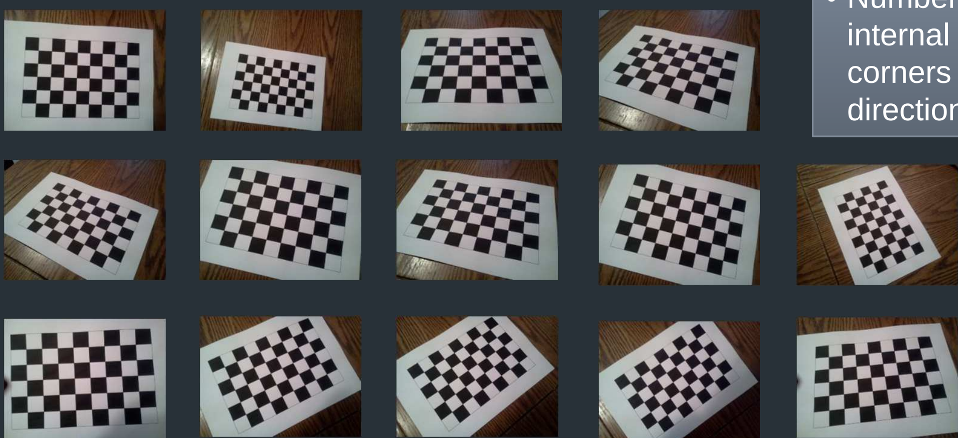


CAMERA CALIBRATION

Fully automated approach by taking pictures of a planar chessboard pattern shown at different orientations, based on *Zhang*'s (*2000*) and *Bouguet*'s (*2013*) methods.

– Calibration of the camera of a laptop computer

Initial Data:
- Chessboard images
- Number of the internal chessboard corners in the two directions

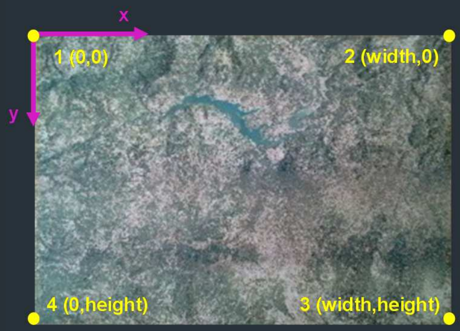


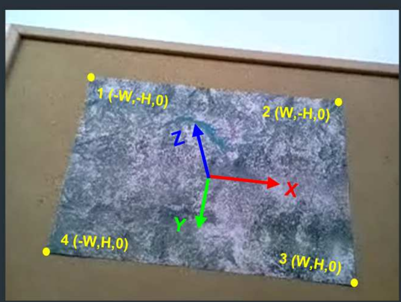
METHODOLOGY OF CAMERA CALIBRATION

- Initial processing of each image
- Check whether the chessboard pattern can be recognized in each image
- Detection of the internal chessboard corners if that check is positive
- Computation of the object coordinates of the internal corners of the chessboard
- Estimation of the initial camera interior orientation
- Computation of the approximate camera exterior orientation for each image, if the chessboard pattern is detected
- Final computation of the camera interior parameters and the camera exterior parameters for each image, using the Levenberg-Marquardt optimization algorithm (*Levenberg,1944*; *Marquardt, 1963*), for the minimization of the reprojection error.

DEFINITION OF THE COORDINATES OF THE PATTERN OBJECT

- Origin: the center of the orthoimage
- X and Y coordinates are derived from the normalized width and height of the pattern image → within the range of [-1, 1]
- Z =0





$$W = \frac{width}{\max(width, height)} \quad \text{και} \quad H = \frac{height}{\max(width, height)}$$

FEATURES EXTRACTION

Features extraction in the pattern image and in every video frame

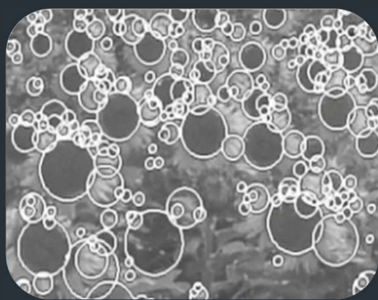
⬇

SURF: Speeded-Up Robust Features (*Bay et al., 2006*)

- Scale invariance
- Rotation invariance
- Skew
- Anisotropic scaling
- Perspective effects

2 steps	**1. Features detection** **2. Features description**

Detection of interest points located in blob-like structures of the image



METHODOLOGY OF SURF (1/2)

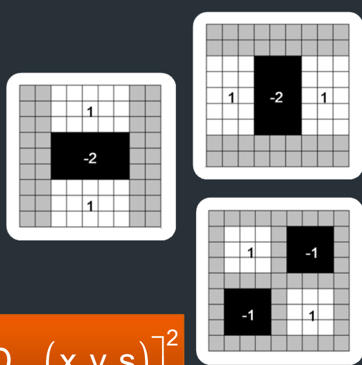
1. Features Detection

Based on the determinant of an approximation of the Hessian Matrix, which is computed for every pixel of the image in all scale levels of each octave into which scale space is divided

$$H_{approx.}(x,y,s) = \begin{bmatrix} D_{xx}(x,y,s) & D_{xy}(x,y,s) \\ D_{xy}(x,y,s) & D_{yy}(x,y,s) \end{bmatrix}$$

determinant

D_{xx}, D_{xy} και D_{yy}: the convolution of box filters → with the image at point (x,y)



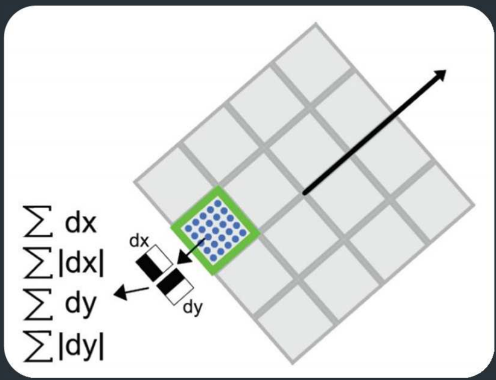
$$\det(H_{approx}) = D_{xx}(x,y,s) \cdot D_{yy}(x,y,s) - \left[w \cdot D_{xy}(x,y,s) \right]^2$$

METHODOLOGY OF SURF (2/2)

2. Features Description

Computation of a 64-dimensional vector for each interest point → descriptor

Indicates the underlying intensity structure of a **square region** around the interest point, oriented along its **dominant orientation**



MATCHING (1/2)

Matching criterion: Euclidean distance between the descriptors of the feature points

❖ Minimization of the outliers → **cross-check test**

Two feature points i and j are matched if:

- the nearest neighbor of the descriptor of point i in the pattern image is the descriptor of point j in the video frame

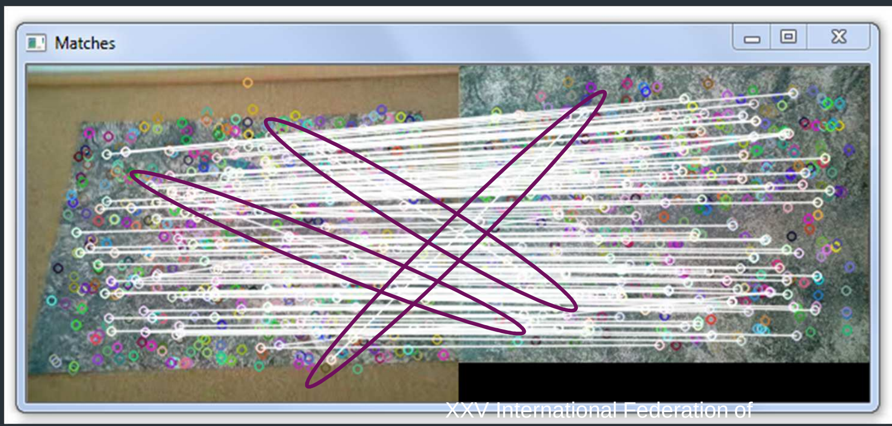
AND

- the nearest neighbor of the descriptor of point j in the video frame is the descriptor of point i in the pattern image

MATCHING (2/2)

Many outliers still remain!

- Definition of a maximum accepted Euclidean distance
- The correspondences are rejected if the Euclidean distance between the descriptors of the matched features points is above that threshold



result

A few incorrect matches are not detected

Final outliers rejection via **RANSAC**

RANSAC FOR THE REJECTION OF OUTLIERS

RANSAC: RANndom SAmple Consensus (*Fischler & Bolles, 1981*)

- **Generally**: Computation of the parameters of a mathematical model using a data set, which may contain many errors, relying on the use of the minimum number of data.
- **In the application**: Rejection of the outliers via the computation of the 2D homography between the pattern image and each video frame

$$s \cdot \begin{bmatrix} x_{frame} \\ y_{frame} \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} x_{pattern} \\ y_{pattern} \\ 1 \end{bmatrix}$$

$$h_{33} = 1$$

RANSAC is applied if **at least 5 matches** are detected.
Otherwise: the orthoimage is not recognized in the frame → the scene is not augmented

ITERATIVE PROCEDURE FOLLOWED BY RANSAC

- A sample of 4 matches is randomly chosen from all matches
- The homography is estimated using the random sample
- The number of valid matches (inliers) is calculated for the above solution
- 3 cases:
  - inliers ≥ threshold → the model is accepted and the algorithm terminates with success
  - inliers < threshold &
    number of iterations = maximum number of iterations →
    the algorithm terminates with failure
  - inliers < threshold &
    number of iterations < maximum number of iterations →
    these steps are repeated

HOMOGRAPHY ESTIMATION

Initial homography estimation via RANSAC using the best set of four matches

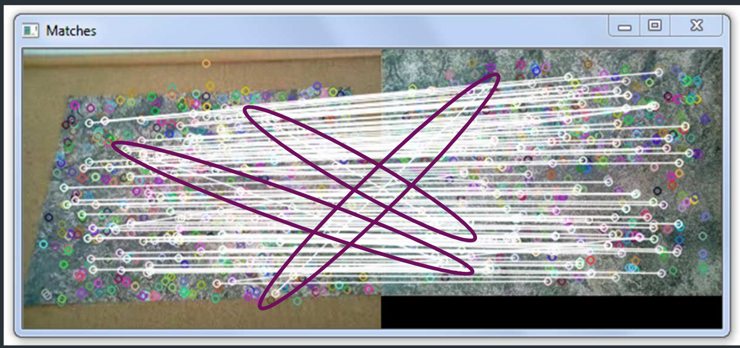
↓

It is refined using the set of all inliers via a nonlinear optimization using the Levenberg-Marquardt algorithm

If **inliers ≥ 5**: the orthoimage is detected in the video frame
Otherwise: the scene is not augmented

RESULT

OUTLIERS REJECTION VIA RANSAC



Matches before the rejection of the outliers by RANSAC

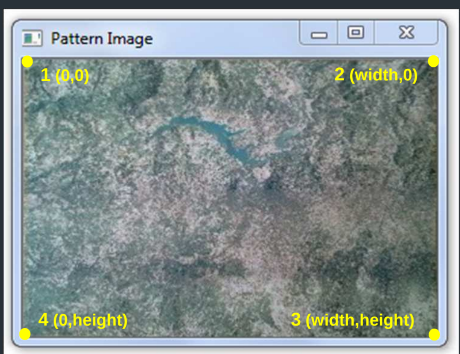
Inliers returned by RANSAC

PATTERN RECOGNITION

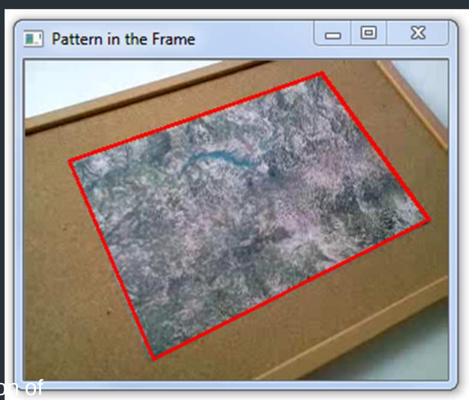
- Computation of the pixel coordinates of the four corners of the pattern object in the video frame

$$\begin{bmatrix} x'_{frame} \\ y'_{frame} \\ w'_{frame} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{pattern} \\ y_{pattern} \\ 1 \end{bmatrix}$$

$$x_{frame} = x'_{frame} / w'_{frame}$$
$$y_{frame} = y'_{frame} / w'_{frame}$$



Homography Matrix



CAMERA EXTERIOR ORIENTATION ESTIMATION (1/4)

Camera interior orientation

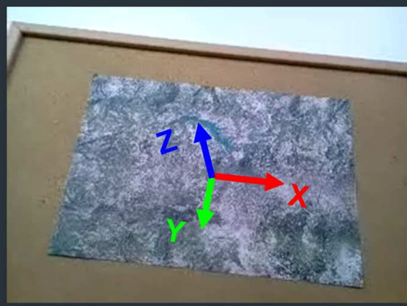
Pixel coordinates of the four corners of the pattern object in the video frame

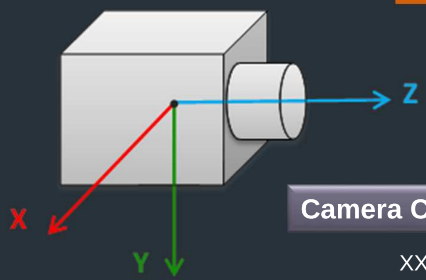
Object coordinates of the four corners of the pattern object

→ Rotation

Translation

Object Coordinate System





Camera Coordinate System

CAMERA EXTERIOR ORIENTATION ESTIMATION (2/4)

Mathematical model: Projection transformation

$$\lambda \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} c_x & 0 & x_0 \\ 0 & c_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera matrix **K**

Joint rotation-translation matrix [**R**|**t**]

Homogeneous coordinates

c_x=c/width$_{pixel}$, c_y=c/height$_{pixel}$

x_0, y_0: pixel coordinates of the principal point

s: skewness

r_{ij}: elements of the rotation matrix

t_i: elements of the translation vector

x, y: pixel coordinates

X, Y, Z: ground coordinates

CAMERA EXTERIOR ORIENTATION ESTIMATION (3/4)

- Linear computation of the approximate elements of the joint rotation-translation matrix using:
  - the homography that relates the planar object coordinates and the pixel coordinates of the corners of the orthoimage after their undistortion
  - the camera interior orientation

$$[R \mid t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \end{bmatrix} \rightarrow r_1 \quad r_2 \quad r_3 \quad t$$

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \rightarrow h_1 \quad h_2 \quad h_3$$

$$r_1 = \lambda \cdot K^{-1} \cdot h_1$$

$$r_2 = \lambda \cdot K^{-1} \cdot h_2$$

$$r_3 = r_1 \times r_2$$

$$\lambda = \frac{1}{\left\| K^{-1} \cdot h_1 \right\|}$$

$$t = \lambda \cdot K^{-1} \cdot h_3$$

CAMERA EXTERIOR ORIENTATION ESTIMATION (4/4)

- Calculation of the Singular Value Decomposition (SVD) of the rotation matrix **R** and recalculation of **R** in order to satisfy the orthogonality condition

$$R = U \cdot W \cdot V^T \xrightarrow{W = I} R = U \cdot V^T$$

- Conversion of **R** into a 3D rotation vector via Rodrigues formula
  - Parallel to the rotation axis
  - Magnitude equal to the magnitude of the rotation

- Levenberg-Marquardt optimization in order to refine the translation and rotation vectors

RENDERING OF THE AUGMENTED SCENE (1/4)

2 steps:

1. Rendering of the video frame on a computer window, so that it forms its background
2. Rendering of the DTM on that window

The coordinates of the vertices of the DTM are defined in a local coordinate system ≠ object coordinate system → They are transformed into the object coordinate system by being normalized into the range of [-1,1] → The normalized object coordinates are transformed into the camera system (viewing transformation)

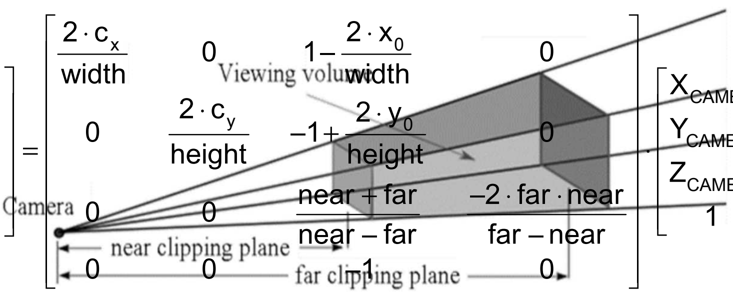
$$\begin{bmatrix} X_{CAMERA} \\ Y_{CAMERA} \\ Z_{CAMERA} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{MODEL} \\ Y_{MODEL} \\ Z_{MODEL} \\ 1 \end{bmatrix}$$

RENDERING OF THE AUGMENTED SCENE (2/4)

Definition of the viewing volume → It determines how the DTM is projected into the final scene + It determines which of its parts are clipped, so as not to be drawn in the final scene

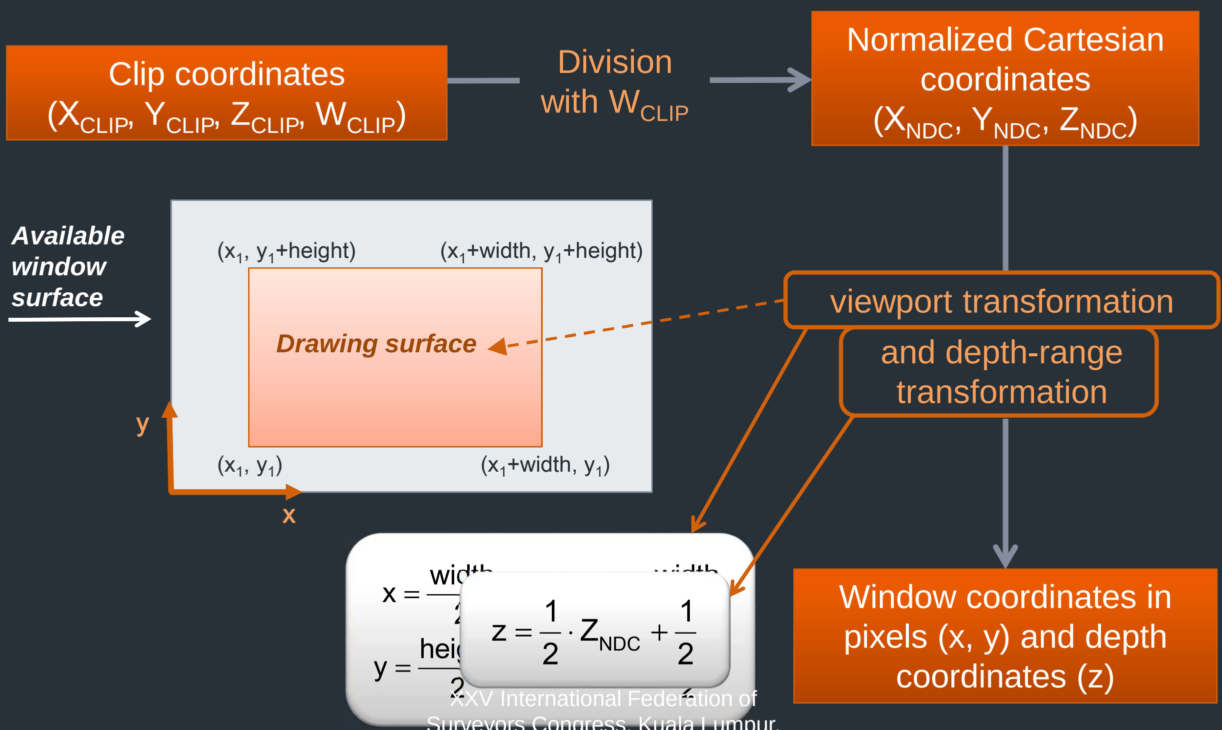
Perspective Projection

Camera coordinates are transformed to clip coordinates



$$\begin{bmatrix} X_{CLIP} \\ Y_{CLIP} \\ Z_{CLIP} \\ W_{CLIP} \end{bmatrix} = \begin{bmatrix} \frac{2 \cdot c_x}{width} & 0 & 1 - \frac{2 \cdot x_0}{width} & 0 \\ 0 & \frac{2 \cdot c_y}{height} & -1 + \frac{2 \cdot y_0}{height} & 0 \\ 0 & 0 & \frac{near + far}{near - far} & \frac{-2 \cdot far \cdot near}{far - near} \\ 0 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_{CAMERA} \\ Y_{CAMERA} \\ Z_{CAMERA} \\ 1 \end{bmatrix}$$

RENDERING OF THE AUGMENTED SCENE (3/4)



RENDERING OF THE AUGMENTED SCENE (4/4)

The orthoimage is draped on the DTM for a realistic representation of the anaglyph

↓

Texture Mapping

Development of the Application

- Programming language: **C++**
- **OpenCV** library (Open source Computer Vision Library)
- **OpenGL** API (Open Graphics Library)

libraries →
  - ▸ OpenGL Core Library
  - ▸ GLU
  - ▸ Freeglut
  - ▸ GLEW

- **GLM: An Alias Wavefrnt OBJ file Library**

❖ The application is intended for computers running Microsoft Windows

RESULTS

XXV International Federation of
Surveyors Congress, Kuala Lumpur,
Malaysia, 16 – 21 June 2014

Thank you for your attention!