6/9/18

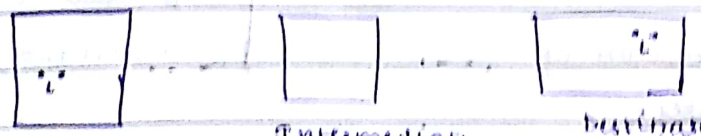
Feature Preserving Morphing

Morphing : One picture $\xrightarrow[\text{smoothly}]{\text{gradually}}$ 2nd picture

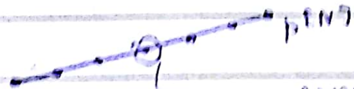
Morphed Image

Features that should be preserved while morphing : FPM
(eg : eyes, nose, ...)

Perform 2 things for FPM :
1) Linearly interpolate co-ordinate values.
2) Linearly interpolate colour values.



Source [#0] . . . Intermediate Image [#K] . . . Destination [#N]



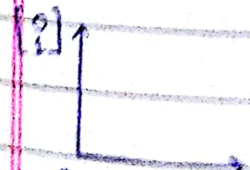
p[0] — p[N]

→ some portion of source & some portion of Dest^n. (No abrupt change)

Warping : Only co-ordinate values changed, no colour blending

→ **Basis** : linearly independent vectors of a vector space & all other vectors of vector space can be formed using linear combination of basis vectors.

eg :



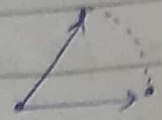
$\begin{bmatrix}0\\1\end{bmatrix}$, $\begin{bmatrix}0\\0\end{bmatrix}$, $\begin{bmatrix}1\\0\end{bmatrix}$

⟶ Unit vectors in x & y - dir^n.

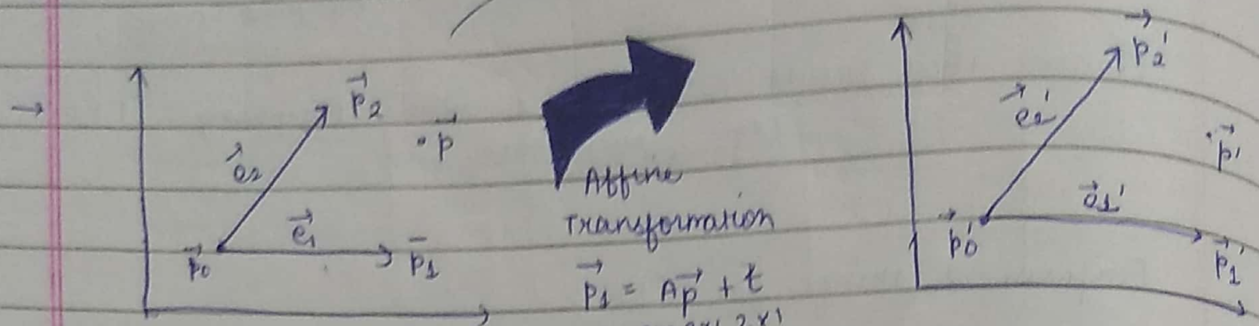
$$\begin{bmatrix}2\\3\end{bmatrix} = 2\begin{bmatrix}1\\0\end{bmatrix} + 3\begin{bmatrix}0\\1\end{bmatrix}$$

Euclidean Space

Affine basis : Any 3 non-collinear points



$\{\vec{p_0}, \vec{p_1}, \vec{p_2}\}$: Affine basis



$$\underset{2\times1}{\vec{p_1}} = \underset{2\times2}{A}\,\underset{2\times1}{\vec{p}} + \underset{2\times1}{t}$$

$\vec{e_1} = \vec{p_1} - \vec{p_0}$

$\vec{e_2} = \vec{p_2} - \vec{p_0}$

$\xrightarrow[\text{prove}]{\text{TO}}$ $\vec{p}' - \vec{p_0}' = \alpha \vec{e_1}' + \beta \vec{e_2}'$

$\boxed{\alpha, \beta \text{ will remain invariant}}$

According to definition of Affine co-ordinates :

$\vec{p} - \vec{p_0} = \alpha \vec{e_1} + \beta \vec{e_2}$ —— (1)

$\langle \alpha, \beta \rangle \rightarrow$ Affine co-ordinates of point $\vec{p}$

TO prove : $\vec{p}' - \vec{p_0}' = \alpha \vec{e_1}' + \beta \vec{e_2}'$

L.H.S. $\vec{p}' - \vec{p_0}' = (A\vec{p} + t) - (A\vec{p_0} + t)$

$= A(\vec{p} - \vec{p_0})$

From eq^n (1),

$= A(\alpha \vec{e_1} + \beta \vec{e_2})$

$= \alpha A(\vec{p_1} - \vec{p_0}) + \beta A(\vec{p_2} - \vec{p_0})$

$= \alpha (A\vec{p_1} - A\vec{p_0}) + \beta (A\vec{p_2} - A\vec{p_0})$

Adding and subtracting t

$= \alpha [(A\vec{p_1} + t) - (A\vec{p_0} + t)] + \beta [(A\vec{p_2} + t) - [A\vec{p_0} + t]]$

$= \alpha\,[\vec{p_1}' - \vec{p_0}'] + \beta\,[\vec{p_2}' - \vec{p_0}']$

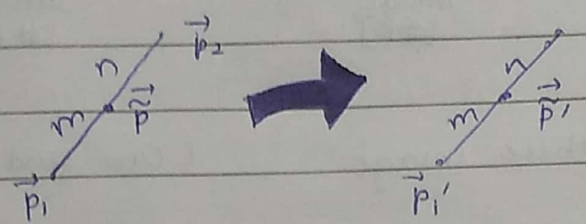
$= \alpha\,\vec{e_1}' + \beta\,\vec{e_2}' = RHS$

$LHS = RHS$

Hence Proved

Ques. Show (for 2-D case) that the division of a line in the ratio $m:n$ is invariant to an affine transformation

Hint :



(Matrix form me easy hoga)

Given : $\dfrac{\vec{p} - \vec{p_1}}{\vec{p_2} - \vec{p}} = \dfrac{m}{n} \Rightarrow n[\vec{p} - \vec{p_1}] - m[\vec{p_2} - \vec{p}] = 0$ —(1)

To prove : $\dfrac{\vec{p}' - \vec{p_1}'}{\vec{p_2}' - \vec{p}'} = \dfrac{m}{n} \Rightarrow n[\vec{p}' - \vec{p_1}'] - m[\vec{p_2}' - \vec{p_0}'] = 0$

L.H.S. : $n[\vec{p}' - \vec{p_1}'] - m[\vec{p_2}' - \vec{p}'] = n[(A\vec{p} + t) - (A\vec{p_1} + t)] - m[(A\vec{p_2} + t) - (A\vec{p} + t)]$

$= n[A\vec{p} - A\vec{p_1}] - m[A\vec{p_2} - A\vec{p}]$

$= A\{\, n[\vec{p} - \vec{p_1}] - m[\vec{p_2} - \vec{p}] \,\}$

= Using (1),

$= A\,[0]$

$= 0 = R.H.S.$

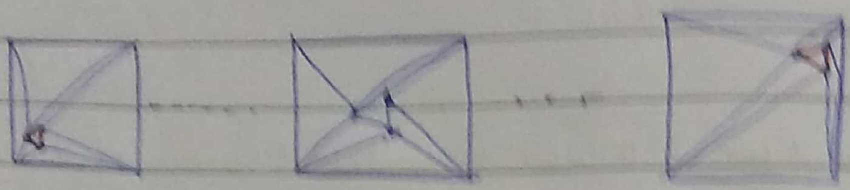
$L.H.S. = R.H.S.$

Hence Proved

→ Multiple ways to do → we are choosing simplest !

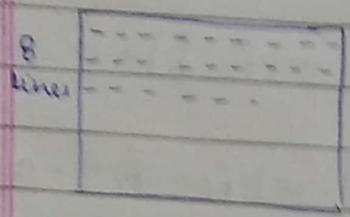
Feature Preserving Morphing (same size images)

↳ To restore Affine transformed image, we need atleast 3 points



Source Image [#0] — Intermediate Image [#k] — Destⁿ Image [#N]

① Triangulate these image (can find co-ordinates manually initially)

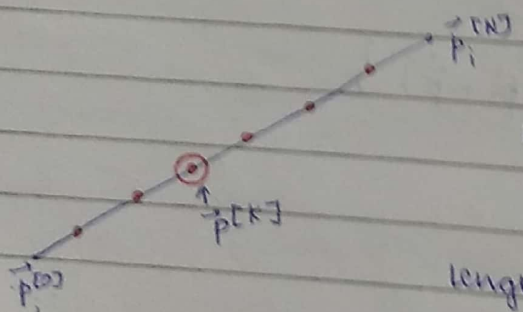


Source code.txt

manually → write co-ordinates of Δ in a file

automatic → Delaunay Triangulation, Voronoi

② Interpolate - coordinates
colour values



N - intervals

(N-1) intermediate points

length of each interval

$$\Delta l = \frac{\vec{p}_i^{[N]} - \vec{p}_i^{[0]}}{N}$$

The position of the i^{th} point in image # k

$$\vec{p}_i^{[k]} = \vec{p}_i^{[0]} + k\Delta l$$

$$= \vec{p}_i^{[0]} + k\left[\frac{\vec{p}_i^{[N]} - \vec{p}_i^{[0]}}{N}\right]$$

$$\vec{p}_i^{[k]} = \left(\frac{N-k}{N}\right)\vec{p}_i^{[0]} + \left(\frac{k}{N}\right)\vec{p}_i^{[N]}$$

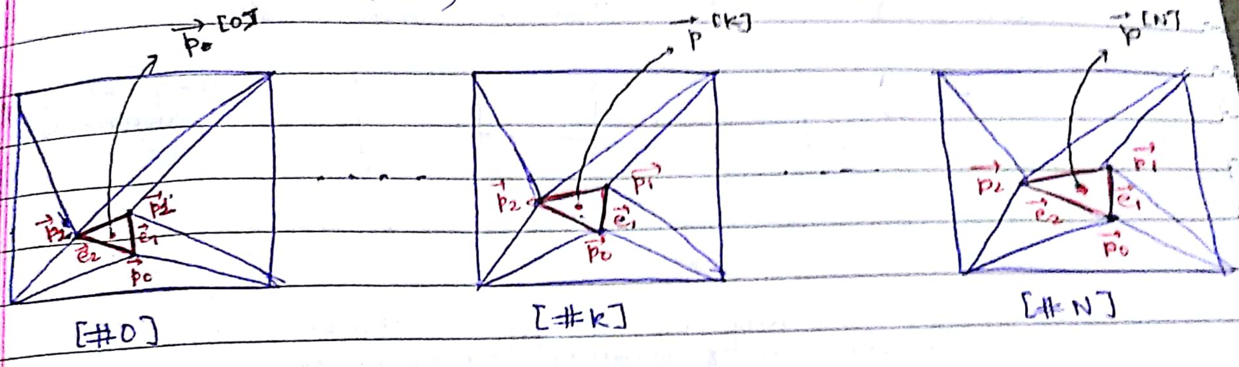
Scanned by CamScanner

15th September' Albert Hall

PAGE NO.:
DATE: / /

Similarly,

$$color(\vec{p}_i^{[k]}) = \left(\frac{N-k}{N}\right) color(\vec{p}_i^{[0]}) + \left(\frac{k}{N}\right) color(\vec{p}_i^{[N]})$$



No. of frames = 10 (More N, more smooth change)

k = 6

apply $\bar{p}^{[k]} - \bar{p}_0^{[k]} = \alpha \bar{e}_1^{[k]} + \beta \bar{e}_2^{[k]}$: can find $\langle \alpha, \beta \rangle$

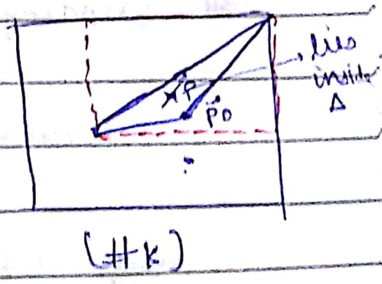
known, k, k, k

can move on whole image & find points which lie in which Δ.

~~scene~~ for source : $\vec{p}^{[0]} - \vec{p}_0^{[0]} = \alpha \vec{e}_1^{[0]} + \beta \vec{e}_2^{[0]}$ (same for destⁿ)

unknown

→ For α & β :



$$\bar{p} - \bar{p}_0 = \alpha \bar{e}_1 + \beta \bar{e}_2$$

$$\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \alpha \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \beta \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

$$\begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix}^{-1} \begin{bmatrix} e \\ f \end{bmatrix}$$

know coordinates of Δ. Traverse & → in red region ~~find~~ find whether a point lies inside Δ or not

Hence, we know $\vec{p} - \vec{p}_0$, $\vec{e}_1$, $\vec{e}_2$

So, we can get values of $\vec{\alpha}$ & $\vec{\beta}$

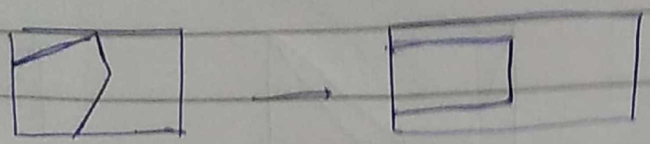
Teacher's Signature..............

11/09/18

DATE: / /

Image Mosaicing

→ Quiz : /\ Projective Trans^n Allign 2 lines



→ apply Reverse Transformation

Panorama : Merging of images (multiple)
↳ maintaining planarity (not moving camera while taking picture)

→ Planar Image Mosaics ✓

App^n (Mosaics) :

↳ To ↑ my view ~~not~~ without adding extra costs.

3 things in Image Mosaics :

1.) Image Alignment : Take 2 images ⇒ Find 4 common pts for alignment & apply ~~perspective~~ projective transf^n.

2) Cut and Paste

3) Blending

Observation : Brightness issue, can't be handled

Image 1 & 2 → get O/p + 3 → ----

Image Mosaics :

1) To increase the field of view without increasing cost.

Two images of a planar object are related by Homography

$$\underset{3\times1}{\vec{p}'} = \underset{3\times3}{H}\,\underset{3\times1}{\vec{p}}$$

PAGE NO. :
DATE : / /

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$ uka proof !

best way, not always necessary ↳ 1

→ If not moving centre of camera, 2 images we obtain are planar.

Proof : Object : 3-D and its image : 2-D

Projective Camera Eq^n : for 3-D image (Homogenous system → 4 vectors)

point in 2-D image, camera matrix

$$\underset{1\times1 \text{ (scalar)}}{\lambda} \underset{3\times1}{\vec{p}} = \underset{3\times4}{M} \underset{4\times1}{\vec{P_w}} \quad \text{--- (A)}$$

~~Multiply both sides by M^* [left inverse of M]~~

$$\underset{4\times1}{\vec{P_w}} = \underset{1\times1}{\lambda} \underset{4\times3}{M^*} \underset{3\times1}{\vec{p}} + \underset{1\times1 \text{ (scalar)}}{s} \underset{4\times1}{\vec{n}} \quad \text{--- (1)}$$

(because M is not square matrix, we get $s\vec{n}$)

$M^* \rightarrow$ left inverse of M

$n \rightarrow$ ~~null sp~~ is in the null space of $\underset{3\times4}{M}$

(A) : $$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

1426
4360

The equation of a plane in world co-ordinates can be written as :

$$ax + by + cz + dH = 0$$ → * Only true in case of planar objects.

$$\boxed{\bar{C} \cdot \bar{P_w} = 0} \quad \text{--- (2)}$$

$\bar{C}$: vector of coefficients

PAGE NO.:
DATE: / /

By putting eq^n (2) in (1),

$$\vec{P_w} = M^* \bar{p} + s\vec{n} \quad \text{—— (1)}$$

$$= \bar{C}^T M^* \bar{p} + s\bar{C}^T \bar{n} = 0$$

$$\Rightarrow \boxed{s = -\frac{\bar{C}^T M^* \bar{p}}{\bar{C}^T \bar{n}}} \quad \text{—— (3)}$$

Putting (3) back in eq^n (1),

$$\bar{P_w} = \lambda M^* \bar{p} + s\bar{n} \quad \text{—— (1)}$$

$$\bar{P_w} = \lambda M^* \bar{p} - \bar{n} \left(\frac{\bar{C}^T_{1\times4} \, M^*_{4\times3} \, \bar{p}_{3\times1}}{\bar{C}^T_{1\times4} \, \bar{n}_{4\times1}} \right)$$

Taking $M^* \bar{p}$ common, $\underset{4\times3}{M^*} \underset{3\times4}{M} = 4\times4$

$$\bar{P}_{w_{4\times1}} = \underbrace{\left(I_{4\times4} - \frac{\bar{n}_{4\times1} \, \bar{C}^T_{1\times4}}{\bar{C}^T_{1\times4} \, \bar{n}_{4\times1}} \right)}_{\text{Scalar}} M^*_{4\times3} \, \bar{P}_{3\times1}$$

$$\boxed{\bar{P}_{w_{4\times1}} = \tilde{M}_{4\times3} \, \bar{p}_{3\times1}} \quad \text{—— (4)}$$

This is for 1 view

Another view point

any object we want to capture. point of that object

world point will be same

camera projection point will be diff. (also, point in image " " diff.)

$$\underset{3\times1}{\bar{p}'} = \underset{3\times4}{M'} \, \underset{4\times1}{\bar{P_w}}$$

* Centre of camera is fixed.

Plane : Distance b/w camera & object should be same

PAGE NO.:
DATE: / /

using eqⁿ (4),

$$\bar{p}' = \underbrace{\underset{3\times4}{M'}\ \underset{4\times3}{\tilde{M}}}_{3\times3}\ \underset{3\times1}{\bar{p}}$$

↓

Homography Matrix (H)

Hence, if object is planar, images are related by Homography.

Hence Proved.

Assignment :-

* can blend only when planar objects are taken into account (even when there are common points in non-planar objects, we can't blend them).

Check : Try to take point from wood. "Inconsistency" (small) will be there. won't get smooth image.

•) will be affective in case of taking scenary photos (all images are nearly at ∞)

Blackboard image mosaics

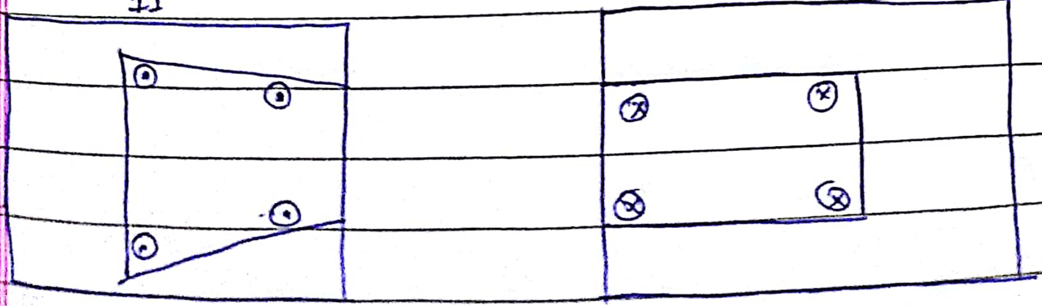
Hence, no affine transfⁿ applied

need 4 points (not 3 because no more parallel property is invariant)

Try to take farthest points so as to get largest common area possible.

don't use : find homograph

I1 I2



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\Rightarrow \frac{x'}{1} = \frac{xh_{11} + yh_{12} + h_{13}}{xh_{31} + yh_{32} + h_{33}}$$

$$\frac{y'}{1} = \frac{xh_{21} + yh_{22} + h_{23}}{xh_{31} + yh_{32} + h_{33}}$$

eg.

$$\begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} 5 \\ 9 \\ 1 \end{bmatrix}$$

$$\rightarrow 10h_{31} + 18h_{32} + 2 = 5h_{11} + 9h_{12} + h_{13}$$

$$20h_{21} + 36h_{32} + 4 = 5h_{21} + 9h_{22} + h_{23}$$

From 1 corresponding pt pair → got 2 eqns

4 → 8 eqns → can find H matrix in this way.

Representation

$$\begin{bmatrix} 5 & 9 & 1 & 0 & 0 & 0 & -10 & -18 & -2 \\ 0 & 0 & 0 & 5 & 9 & 1 & -20 & -36 & -4 \\ & & & & \vdots & & & & \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ 1 \end{bmatrix}$$

↓ 8 × 9 matrix

↓ 9 × 1

9th October

Design your own Product → OpenCV.

PAGE NO.:
DATE: / /

$A\vec{h} = \vec{0}$ → can be solved using SVD (can use in-built func^n)

↓

will get 3 matrices

U, S, V^T

↓

last column of V^T will give H.

$$\underset{M \times N}{A} = \underset{M \times M}{U}\ \underset{M \times N}{S}\ \underset{N \times N}{V^T}$$

Eg. $8 \times 9 \quad 8 \times 8 \quad 8 \times 9 \quad 9 \times 9$

↓

~~last column of V~~

(9×1) → gives H

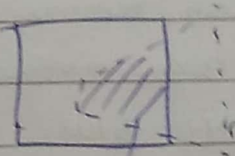
&

can get corners of 2^{nd} image & blend common part of 1^{st} & 2^{nd} image.

~~1st image~~

Then, we can do reverse mapping

→ Blending : can take pixel either from 1^{st} or 2^{nd} image



Blend this region

OpenCV.

```
                           C     R
Mat m ( size (400, 200), CV_ 8UC3) ;
Mat subImg ( m , Rect (10, 10
```