Project Title

Awesome Typography: Statistics-Based Text Effects Transfer

Team members:

Samartha (2018101094)-CSE
Abhiram (2018102036)-ECE
Nihar (2018102039) -ECE
Tushar (2018102041)-ECE

Mentor TA :

Surendra Kumar

Team 31:

255 Shades of Gray

Repo URL :

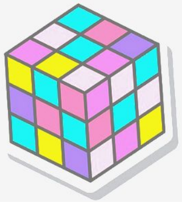
https://github.com/Digital-Image-Processing-IIITH/project-255-shades-of-gray

Introduction...

It is often a good practice in Publishing and advertising to use special effects to the text to make it more attractive. But this task is very cumbersome as it requires experienced designers and advanced editing skills, and also a lot of time for getting a visually plausible output.

The aim of the paper is to generate fantastic special effects for typography by using few Image processing Techniques, which will be discussed in the subsequent slides.

Problem Statement

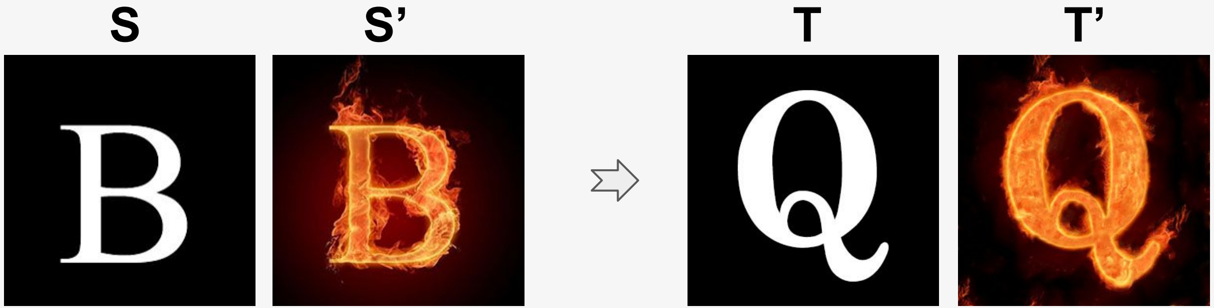


Text effects transfer takes a set of three images as input :

- Raw Source Image, *S*
- Source Stylised Image *S’*
- Raw Target text image *T*

Now, we should generate target stylised image such that *S:S’::T:T’*

We deal this problem in 4 steps, let us see how ...



Procedure

The paper divides style transfer into different sections which are summarized in the following slides as follows

- ➢ Patch partitioning
- ➢ Optimal patch scale detection
- ➢ Posterior probability estimation
- ➢ Effect transfer

1. Patch partition

Patch Partitioning

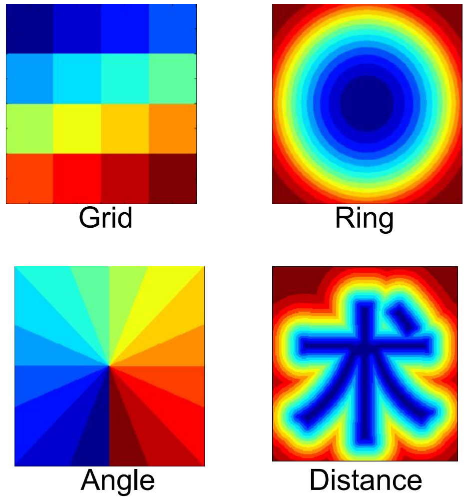
Patch is a segment of image, which is used for style transfer. We have to identify patches first, to use it for style transfer. Generally, patch patterns are highly dominated by their locations.

We identify pattern of a patch by two factors :

- Pixel color
- Patch scale

To quantitatively evaluate the location of patches we divide the image into say, N partitions. The paper discusses about the given partition methods and their effectiveness in partitioning the patches in the stylized images. (Distance here is distance from skeleton)

Different partition modes explored



Patch Partitioning

For finding the best partition method, the researchers have done the following. The training accuracy of an SVM which was trained on the images to classify the the color/scale, given type of partition was measured for a group of images, it turned out that the accuracy is highest for Distance based partitioning. (ref. Table 1 in paper)

r	Random	Grid	Angle	Ring	Distance
Color	0.063	0.106	0.119	0.105	0.147 *
Scale	0.153	0.793	0.486	0.590	0.950 *

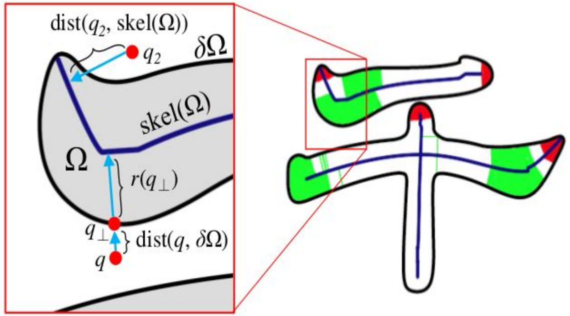
Distance Partition Problems

We find the distance transform from the skeleton to classify the pixel into different patch levels.But there are two problems with this :

- 🌟 Not invariant to text radius
- 🌟 Distances at sharp or flat regions (as in red region of image)

To make the distance width invariant, we normalise the distance of every pixel with distance of nearest point on contour from it (called text radius).

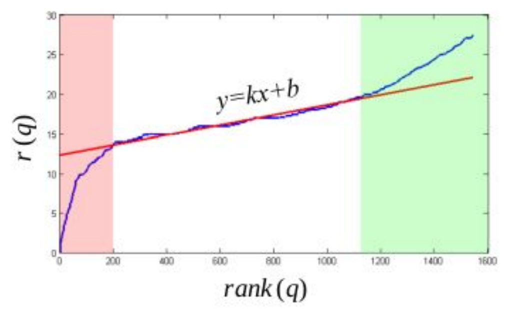
This might give varying results in the sharp or flat regions, instead we normalise with an estimate of text radius.



Problems with general Distance Transform

Distance Partition Modifications

Estimates are found from the regression line. We remove the outliers which cause irregularities (red and green regions in image) we assume them to be leftmost 20% points



(d) Linear relation of $r(q)$ and $rank(q)$

$$\tilde{r}(q) = max(dist(q, skel(\Omega)), 0.2k|\delta\Omega| + b)$$

$$0.2k|\delta\Omega| + b \geqslant dist(q, skel(\Omega)) \; for \; q \in outliers$$

Ω	Pixels in Text Region
$\delta\Omega$	Pixels on Boundary
$\lvert\delta\Omega\rvert$	Boundary pixel count

Distance Partition
Width correction

$$\tilde{dist}(q, skel(\Omega)) = \begin{cases} \frac{1+dist(q,skel(\Omega))}{\tilde{r}}, & \text{if } q \notin Q \\ \frac{1-dist(q,skel(\Omega))}{\tilde{r}(q_\perp)}, & \text{otherwise} \end{cases}$$

Here,

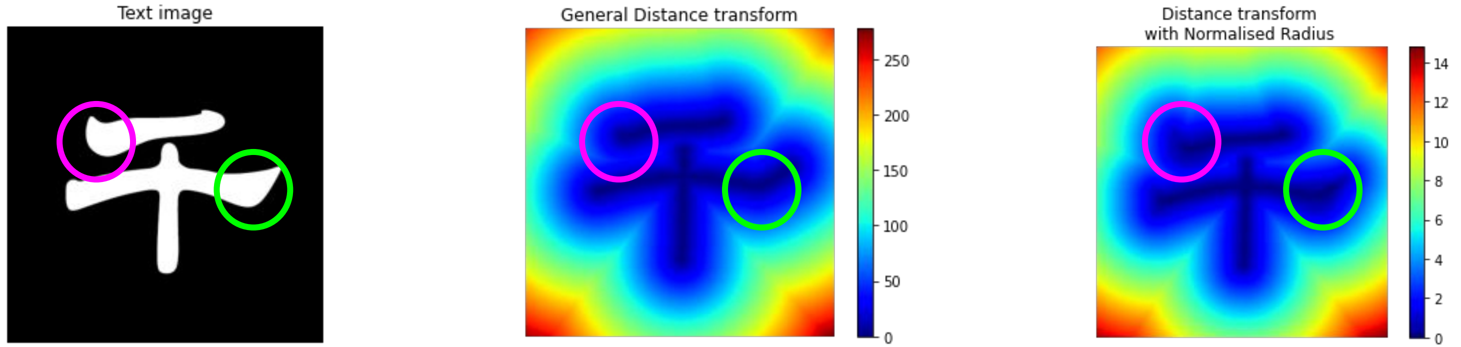
$q_\perp \in \delta\Omega$	Is the nearest pixel to along $\boldsymbol{\delta\Omega}$
$\bar{r} = 0.5k\lvert\delta\Omega\rvert + b$	Is the mean text

Notations,

Ω	Pixels in Text Region
$\delta\Omega$	Pixels on Boundary
$\lvert\delta\Omega\rvert$	Boundary pixel count

$r(q)$	Text width (shortest distance from boundary to given pixel)
$\tilde{r}(q)$	Text width (shortest distance from boundary to given pixel)

Distance Partition
End Result



We can observe the difference between outputs of both transforms, which clearly tell us that Normalised distance better preserves the shape information compared to the general Distance Transform (observe the highlighted regions)

2. Optimal Patch Scale Detection

Optimal Patch Scale Detection

Scale of the patch is the size of the patch that we are using for style transform. We determine the scale of the patch centered at pixel *"q"* by using the following algorithm. (This algorithm acts like a multi-level - coarse to fine sieve)

$$
\begin{aligned}
& R \;=\; \{q \mid q \in S\} \\
& \quad \text{for } l \;=\; L, \ldots, 2 \;: \\
& \quad\quad \text{for all } p \in R \;: \\
& \quad\quad\quad \hat{q} \;=\; \arg\min_{\hat{q}} d_l\,(q, \hat{q}) \\
& \quad\quad\quad \text{if } \zeta_l\,(q, \hat{q}) \text{ is false} \\
& \quad\quad\quad\quad scal\,(q) = l \\
& \quad\quad\quad\quad R = R\;\{q\}
\end{aligned}
$$

Where, $\hat{q}$ is correspondence of q at scale ℓ given by

$$\hat{q} = \text{argmin}(||Q_l(q) - Q_l(\hat{q})||^2 + ||Q_l'(q) - Q_l'(\hat{q})||^2)$$

And ζ_ℓ is the truth value of

$$\zeta_l(q, \hat{q}) = (\sigma_l + \sqrt{d_l(q, \hat{q})} > \omega)$$ where, $$\sigma_1 = \frac{\sqrt{Var(Q_l'(q))}}{2}$$

Optimal Patch Scale Detection

- Where, $\hat{q}$ is correspondence of q at scale ℓ given by

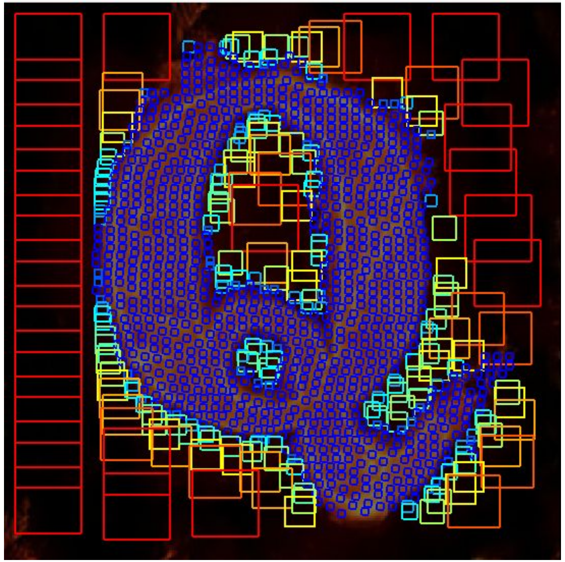
$$\hat{q} = \operatorname{argmin}(||Q_l(q) - Q_l(\hat{q})||^2 + ||Q'_l(q) - Q'_l(\hat{q})||^2)$$

- And ζ_ℓ is the truth value of

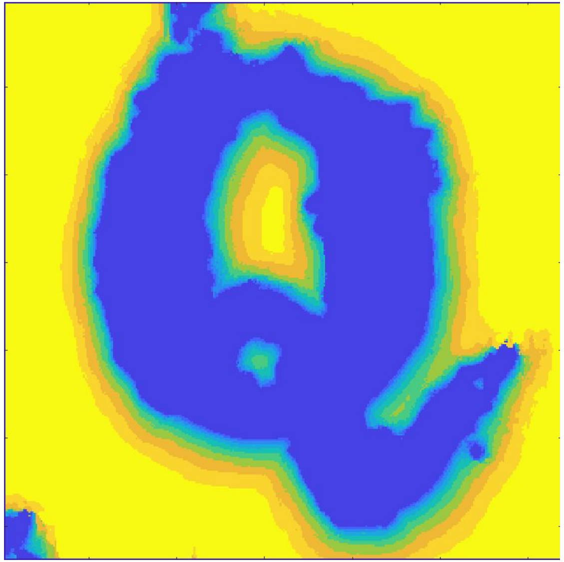
$$\zeta_l(q, \hat{q}) = (\sigma_l + \sqrt{d_l(q, \hat{q})} > \omega) \quad \text{where,} \quad \sigma_1 = \frac{\sqrt{Var(Q'_l(q))}}{2}$$

- $Q'_\ell(q)$ is the patch centered at q at scale ℓ which is done by downsampling the image

Optimal Patch Scale Detection
End Result



Patch Scale Visualised



Optimal Scale Map

3. Optimal scale Posterior probability estimation

Optimal Scale Posterior Probability Estimation

- As we know that there is high correlation between patch patterns and their spatial distributions, we derive posterior probability of optimal patch scale given the partition it belongs to. For this we quantize the distance transform of the image into 100 bins.

- This is done in 3 steps and mathematical expressions are as follows:

Calculating 2D histogram of distance and scale

$$hist(l,x) = \Sigma_q \psi(scale(q) = l \wedge bin(q) = x)$$

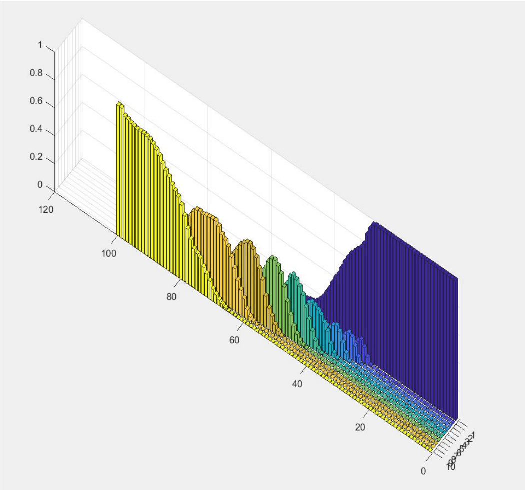
Estimating joint probability of distance and scale

$$P(l,x) = \frac{hist(l,x)}{\Sigma_{l,x} hist(l,x)}$$

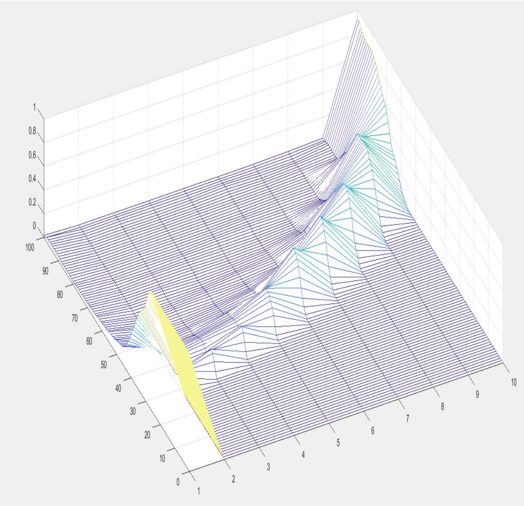
Calculating posterior probability of scale given distance

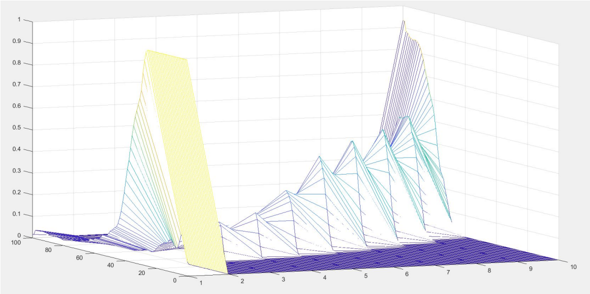
$$P(l|bin(p)) = \frac{P(l,bin(p))}{\Sigma_l \; P(l,bin(p))}$$

Optimal Scale Posterior Probability Estimation



Histogram of distance and scale





Posterior Probability graphs

4. Style Transfer

Style Transfer

Style transfer occurs from patches in source image to patches in target image using an optimization function to perform texture synthesis. Optimization function from paper **[2]** is taken as reference with a few custom modifications. Keep it somewhere :P

We have three criterion for transfering style,
- 🌟 Appearance term
- 🌟 Distribution term
- 🌟 Psychovisual term

This is similar to histogram matching. The objective function is as follows,

$$min_q(\Sigma_p E_{app}(p,q) + \lambda_1 E_{dist}(p,q) + \lambda_2 E_{psy}(p,q))$$

Appearance term
Texture transfer

The main purpose of the appearance term is :

- 🌟 Preserve coarse grained structures
- 🌟 Preserve texture details

$$E_{app}(p,q) = \lambda_3 ||P(p) - Q(q)||^2 + ||P'(p) - Q'(q)||^2$$

$$E_{app}(p,q) = \lambda_3 \Sigma_l P(l|bin(p))||P(p) - Q(q)||^2 + \Sigma_l P(l|bin(p))||P'(p) - Q'(q)||^2$$

Distribution term

Spatial style transfer

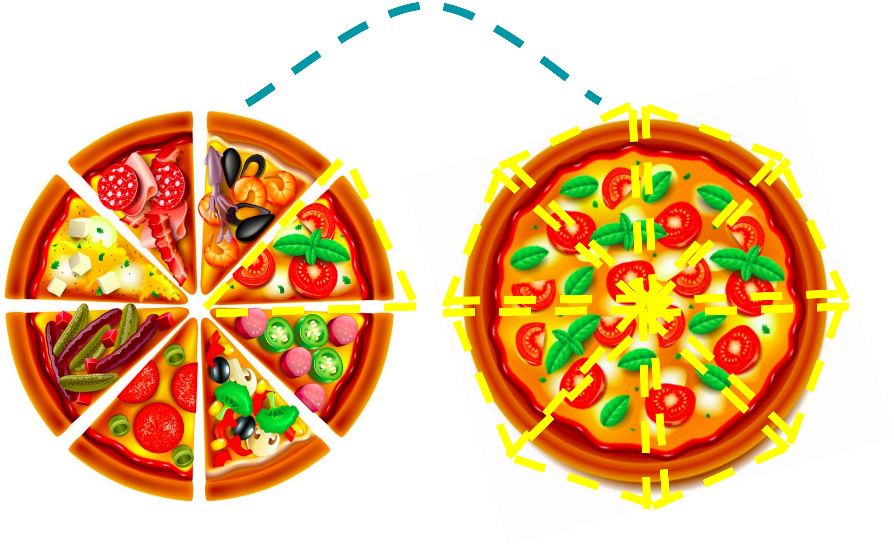
This term ensures that the distribution of sub-effects in the target image and source example are similar, which is the basis for our assumption in Optimal Scale Probability Estimation section that posterior probabilities $p(l|x)$ in T' and S' are same.

$$E_{dist}(p, q) = \frac{(dist(p) - dist(q))^2}{max(1, dist^2(p))}$$

Psychovisual term

Avoiding Repetitiveness

We do not want the same pattern to repeat again and again, so we penalise repetitions. This is done by adding the following term to the objective function.



$$E_{psy}(p, q) = |\phi(q)|$$

$$\Sigma_p |\phi(q)| = \Sigma_q \Sigma_{p\in\phi(q)} |\phi(q)| = \Sigma_q |\phi(q)|^2$$

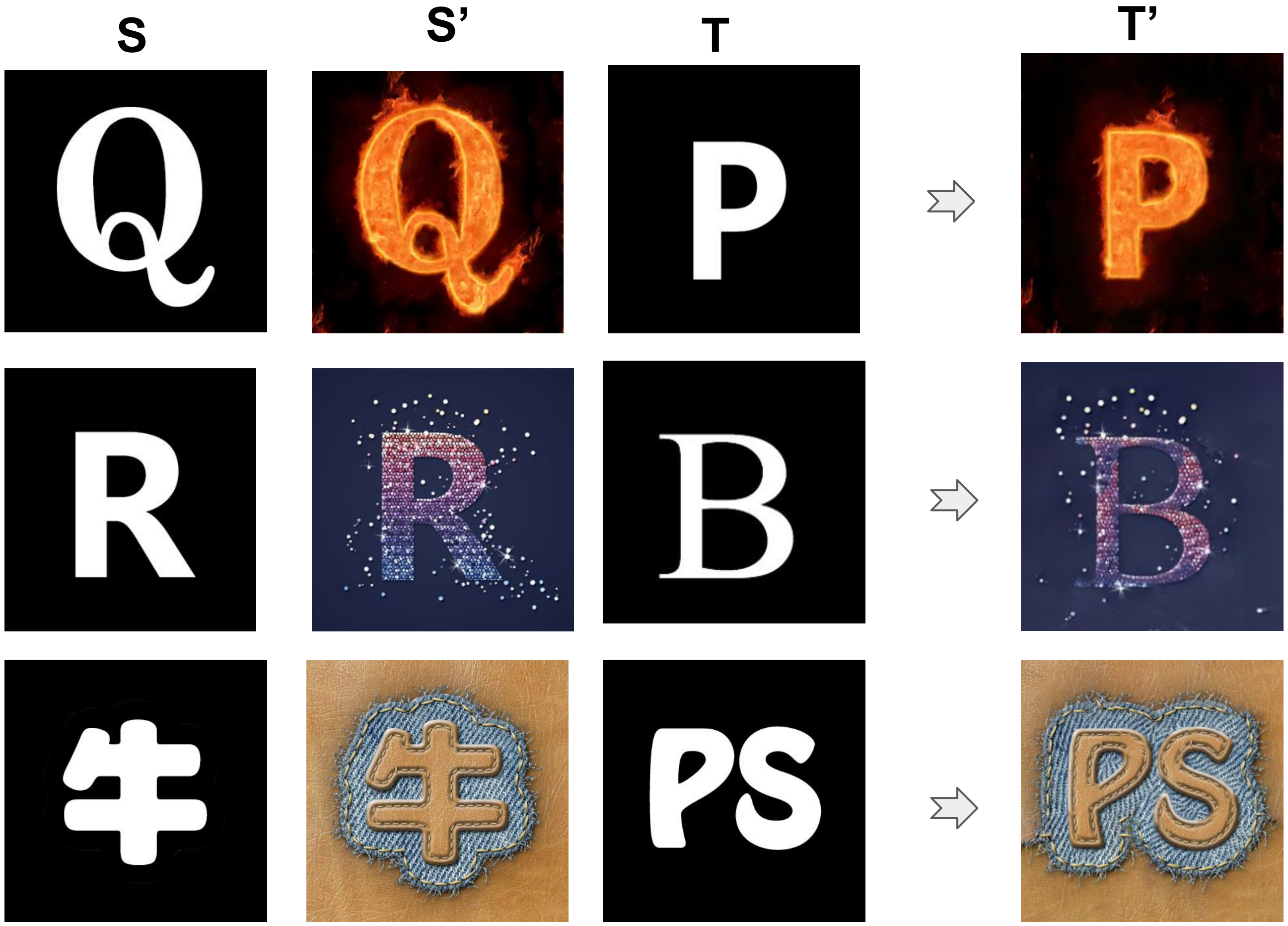
Style Transfer Matching

Nearest Neighbour Field (NNF) mapping algorithm is used to find the patch correspondences. We search with random offset and adjacent offsets search cooperatively. NNF is initialized randomly or using prior information.

Each iteration of the algorithm proceeds as follows: Offsets are examined in scan order (from left to right, top to bottom), and each undergoes propagation followed by random search.

- Propagation - Choosing the offset in the NNF which reduces error. (general iteration step of an optimization problem)
- Random Search - We try to improve the estimate by testing a sequence of candidate offsets which are at an exponentially decreasing distance from the current estimate.

Sample Final Outputs



WORK DIVISION:

Theory :

Section 3.1 : Problem Formulation & Analysis

Lead : Samartha

Section 3.2 : Statistics estimation

3.2.1 : Lead : Samartha

3.2.2 : Lead : Tushar

3.2.3 : Lead : Nihar

Section 3.3 : Style Transfer

Lead : Tushar

Implementation :

Section - 3.2:

3.2.1 : Main Algo : Abhiram

: KNN search : Samartha

3.2.2 : Lead : Tushar

3.2.3 : Lead : Nihar

Section 3.3 :

Lead : Nihar

Presentation :

Lead 1 : Abhiram

Lead 2 : Nihar

Documentation :

Lead 1 : Samartha

Lead 2 : Tushar

Dataset :

Lead : Tushar

References

[1] Shuai Yang, Jiaying Liu, Zhouhui Lian, Zongming Guo; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[2] Y.Wexler, E.Shechtman, and M.Irani. Space-time completion of video. IEEE Transactions on Pattern Analysis and Machine Intelligence, March 2007.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics, August 2009.


A Picture is worth a
Thousand words