Techniques for restoring room images from virtual background images

バーチャル背景適用済み映像から部屋の画像を復元する手法

AV TOKYO

no drink, no hack.

Name：Ryusei Ishikawa（石川 琉聖）

Branch ：Ritsumeikan University

Twitter：@ryusei_ishika

Achievements：

seccamp’19

SecHack365’20

ひとこと：素人質問する前に「僕は玄人です」と言ってください(ToT)

Name：Satoki Tsuji（辻 知希）

Branch：NEET（Nagoya University?）

Twitter：@satoki00

Achievements：

GoogleVRP$5000

CVE...

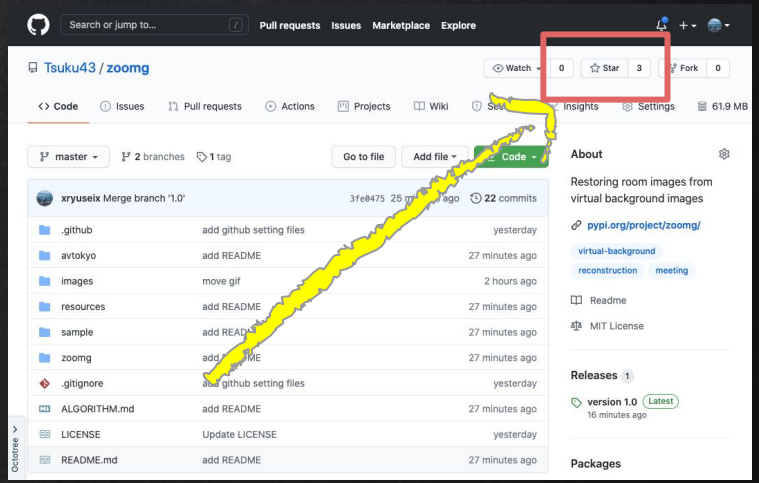
ひとこと：よわよわなのでいぢめないでください(ToT)

RESULT

Zoomg

Zoomg is a tool that extracts color information from the boundary between a virtual background and an object, such as a body, and automatically restores the room hidden behind it.

Zoomg



Latest Development Repositories

https://github.com/Tsuku43/zoomg

MANUALLY

手動で復元するおはなし

Areas used for restoration

Restoration

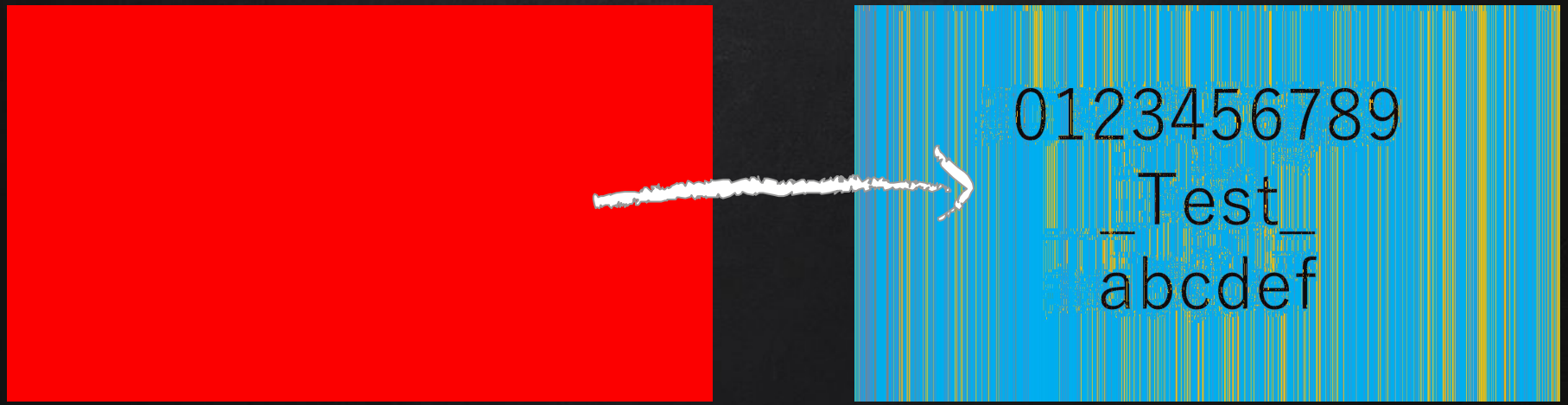
BEFORE

AFTER

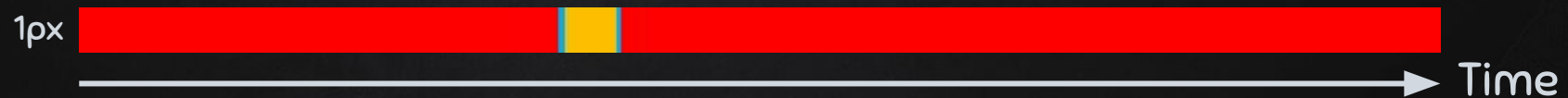
AUTOMATICALLY

自動で復元するおはなし

Idea of restoration



Algorithm



Coding1 - Technology & Language -

Coding2 – Initialization –

Python Script

```python
import zoomg
zoom = zoomg.Zoomg(height, width) # 初期化
while frame:
        frame = frame_read(video) # 画像を取得
        zoom.add_image(frame) # 画像を追加
zoom.generate_image(param) # 画像を生成
image = zoom.get_image() # 生成画像を取得
savefig(image) # 画像を保存
```

1. Creating the array for the restored image

```cpp
using Pixel = std::vector<int>;
using Image = std::vector<vector<Pixel>>;
Image(height, std::vector<Pixel>(width)) image;
```

2. Creation of the array
   for frequency management

```cpp
std::vector<std::vector<
  std::map<Pixel, std::pair<int, int>>>> freq;
// 各ピクセルで平衡二分木を作成する
// std::mapはキーを画素，値は個数と出現タイミングを表す
```

Coding2 - Add Images -

Python Script

```python
import zoomg
zoom = zoomg.Zoomg(height, width) # 初期化
while frame:
    frame = frame_read(video) # 画像を取得
    zoom.add_image(frame) # 画像を追加
zoom.generate_image(param) # 画像を生成
image = zoom.get_image() # 生成画像を取得
savefig(image) # 画像を保存
```

3. Create a tree to manage the frequency of each pixel

```cpp
void add_image(Image img) {
  int order = 0;  // 出現順を保持する
  for (int h = 0; h < height; ++h) {
    for (int w = 0; w < width; ++w) { // 全ての座標について
      Pixel p = img[h][w];
      // ある画素が以前出現したことがある場合
      if (freq[h][w].find(p) != freq[h][w].end()) {
        ++freq[h][w][p].first;
      } else { // ある画素が初めて出現した時
        freq[h][w][p] = {1, order++};
      }
    }
  }}
```

Coding2 - Generate Image -

4. Sort each pixel in order of frequency and timing
5. Search for values cosine similarity less than the param
6. Use that pixel in the reconstructed image

Python Script

```python
import zoomg
zoom = zoomg.Zoomg(height, width) # 初期化
while frame:
    frame = frame_read(video) # 画像を取得
    zoom.add_image(frame) # 画像を追加
zoom.generate_image(param) # 画像を生成
image = zoom.get_image() # 生成画像を取得
savefig(image) # 画像を保存
```

```cpp
// 予めソートした配列 = most_commonを作成しておく
Pixel most = *most_common.rbegin(); // 最頻値を取得
for (auto p : most_common) {
  if (cos_sim(p, most) < param) { // コサイン類似度を計算
    image[h][w] = p; // 復元できた画素として特定
    return;
  }
}
image[h][w] = most; // 復元できなかった場合
```

Coding2 – Getting a generated image –

Python Script

```
import zoomg
zoom = zoomg.Zoomg(height, width) # 初期化
while frame:
      frame = frame_read(video) # 画像を取得
      zoom.add_image(frame) # 画像を追加
zoom.generate_image(param) # 画像を生成
image = zoom.get_image() # 生成画像を取得
savefig(image) # 画像を保存
```

7. Get a restored image

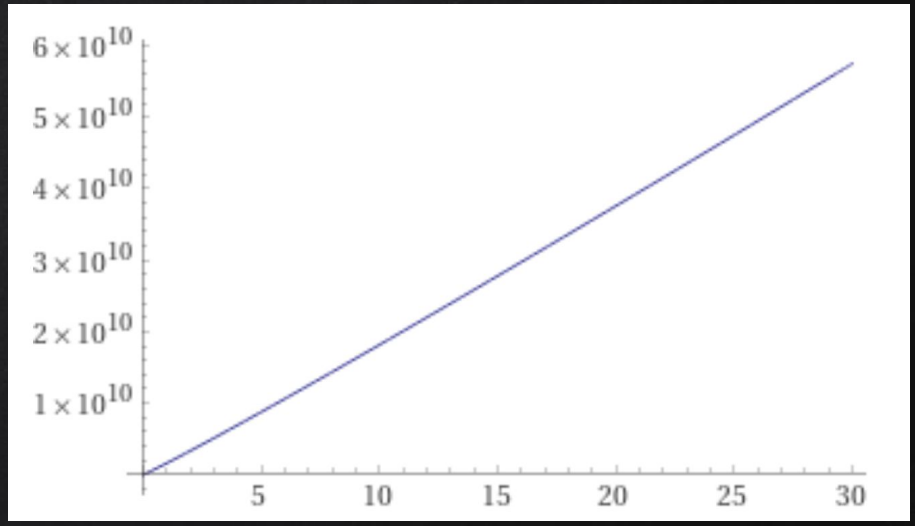
```
return image;
```

Coding3 – Computational cost –

Time complexity

F := frames of the video
H := height of the image
W := width of the image

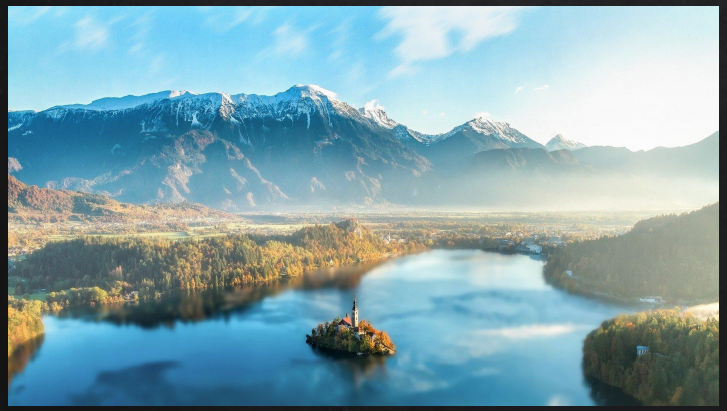
$O(FHW \log_2 (FHW))$



Estimated number of calculations for video playback time
X : Time [sec], Y : Estimated number of operations [times]

Evaluation

BACKGROUND



RECONSTRUCTIO

Defense

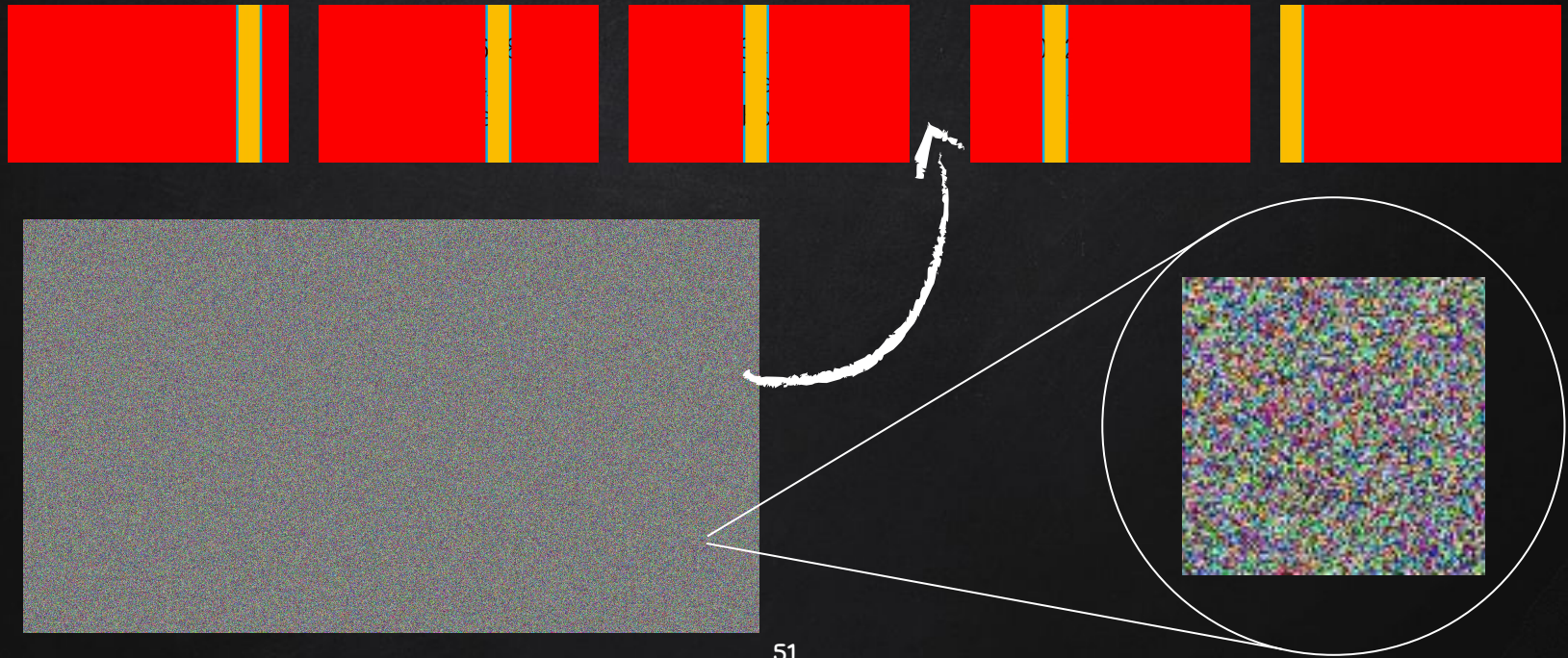
復元を**防御**するおはなし

Idea of defensive techniques

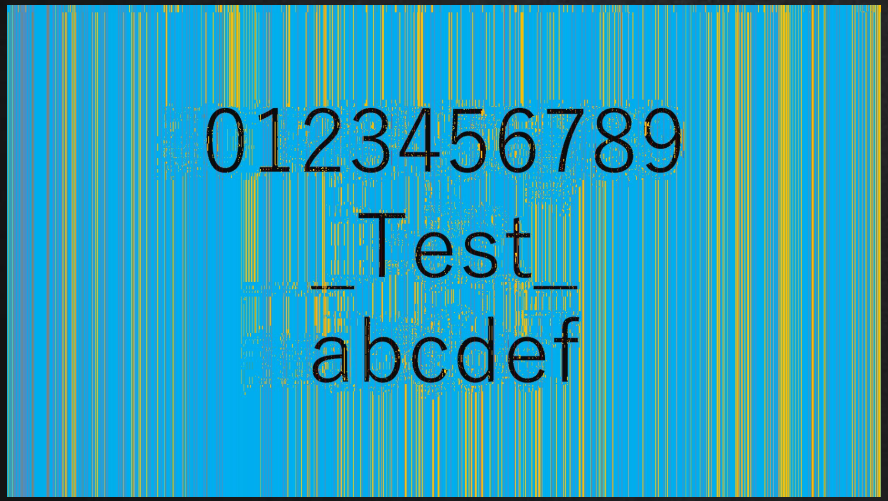
Wouldn't that be better on video?

- ✗ Easily available videos
- ✗ Short, looping videos

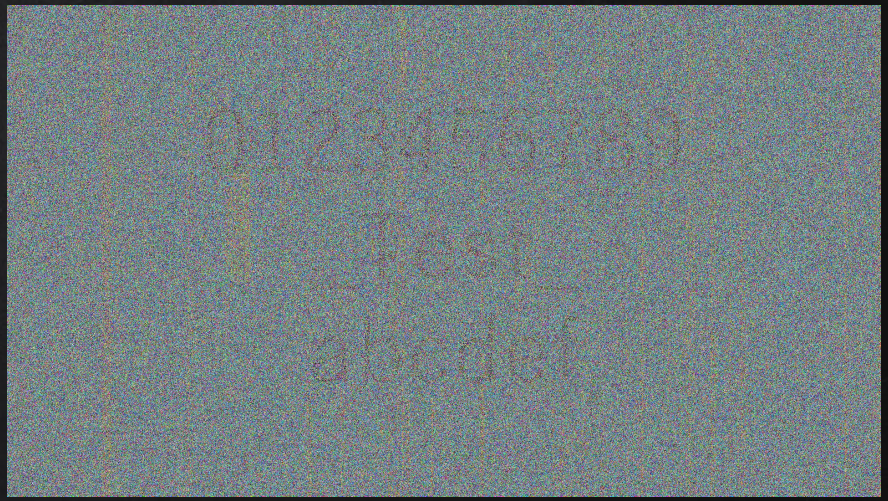
Defense1 – frame noises –



Defense1 – frame noises –

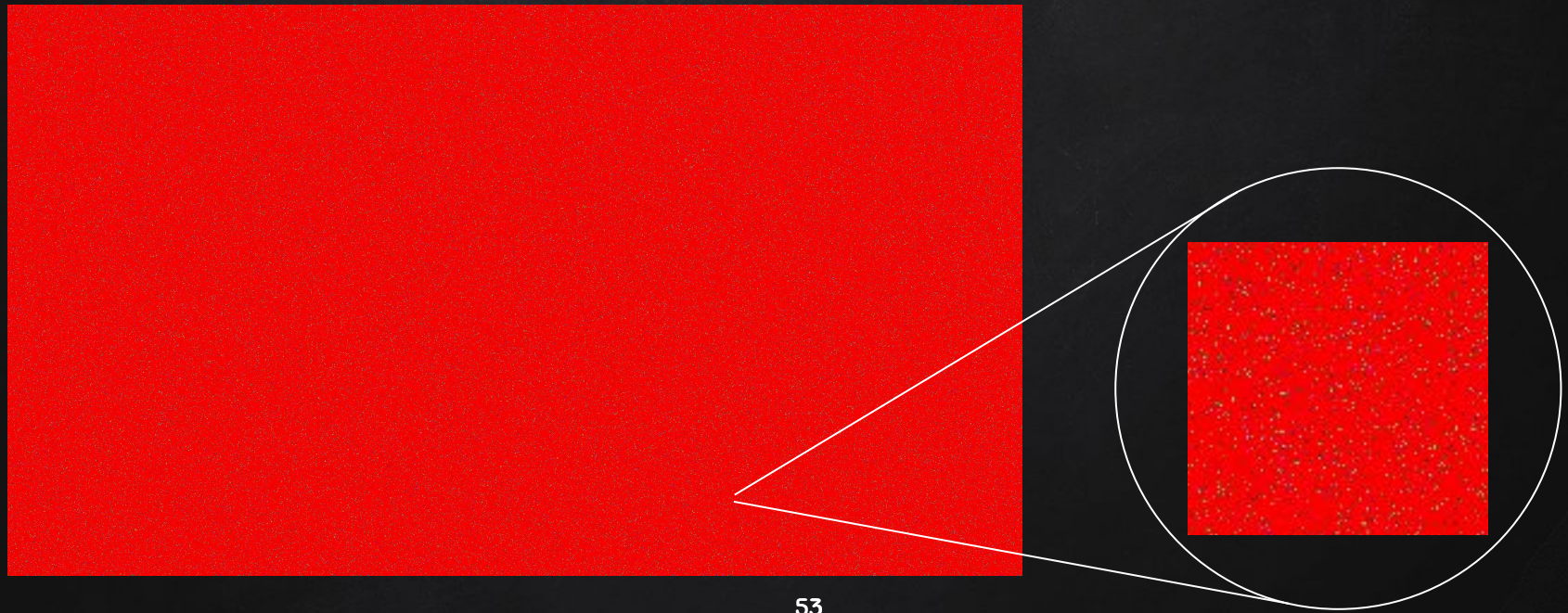


BEFORE

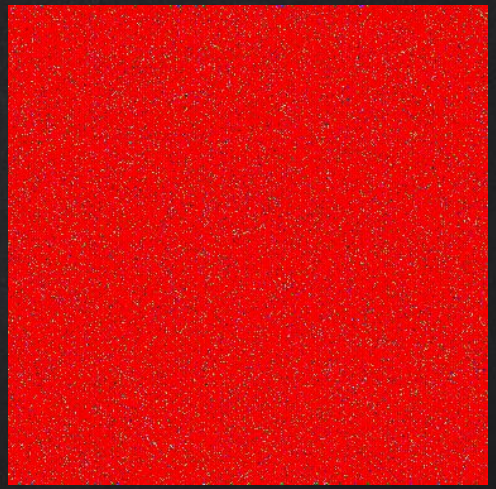


AFTER

Defense2 – pixel noises –



Defense2 – pixel noises –

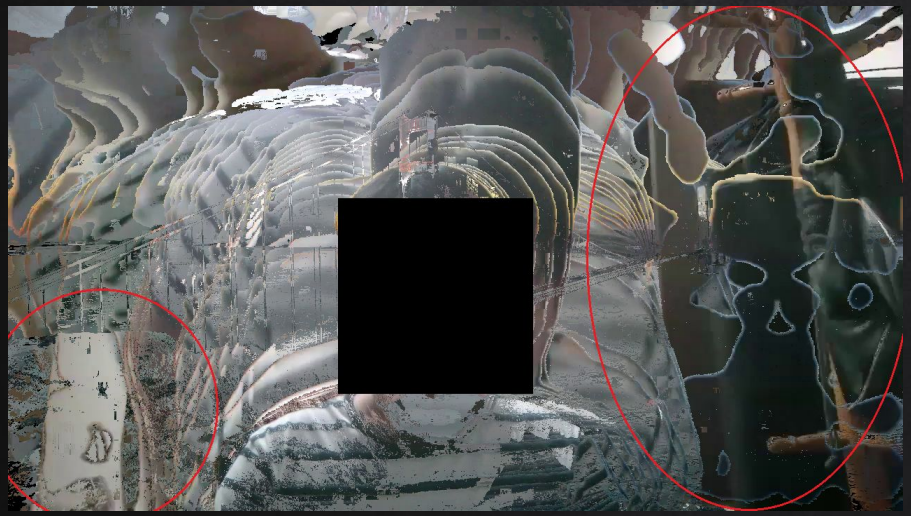


The Best Defense Method

PRACTICALLY

実際のおはなし

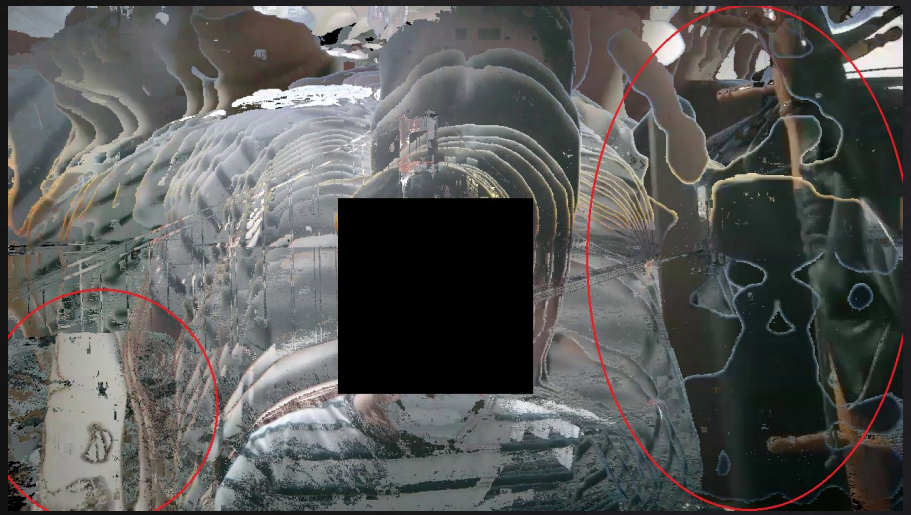
Restoring room image from virtual background images



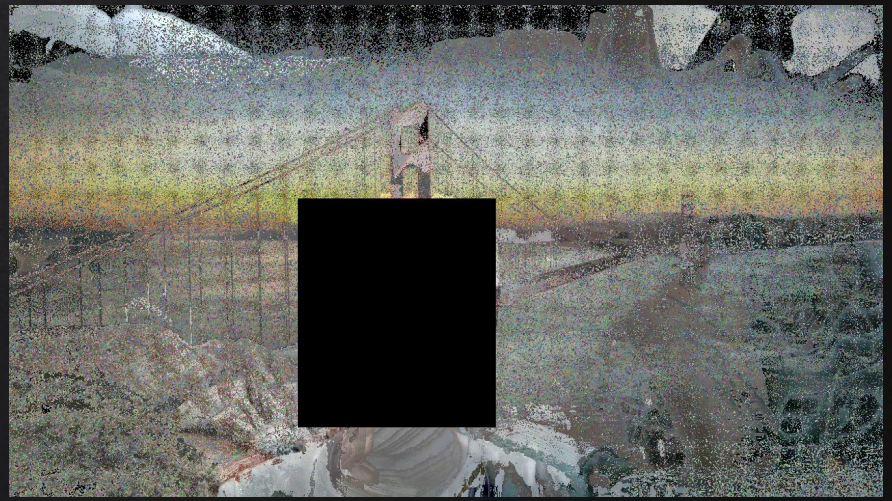
Restored Room

Real Room

Pixel noises defense



Restored without noises



Restored with noises

Capture The Flag !!!

https://bit.ly/37TnyB6