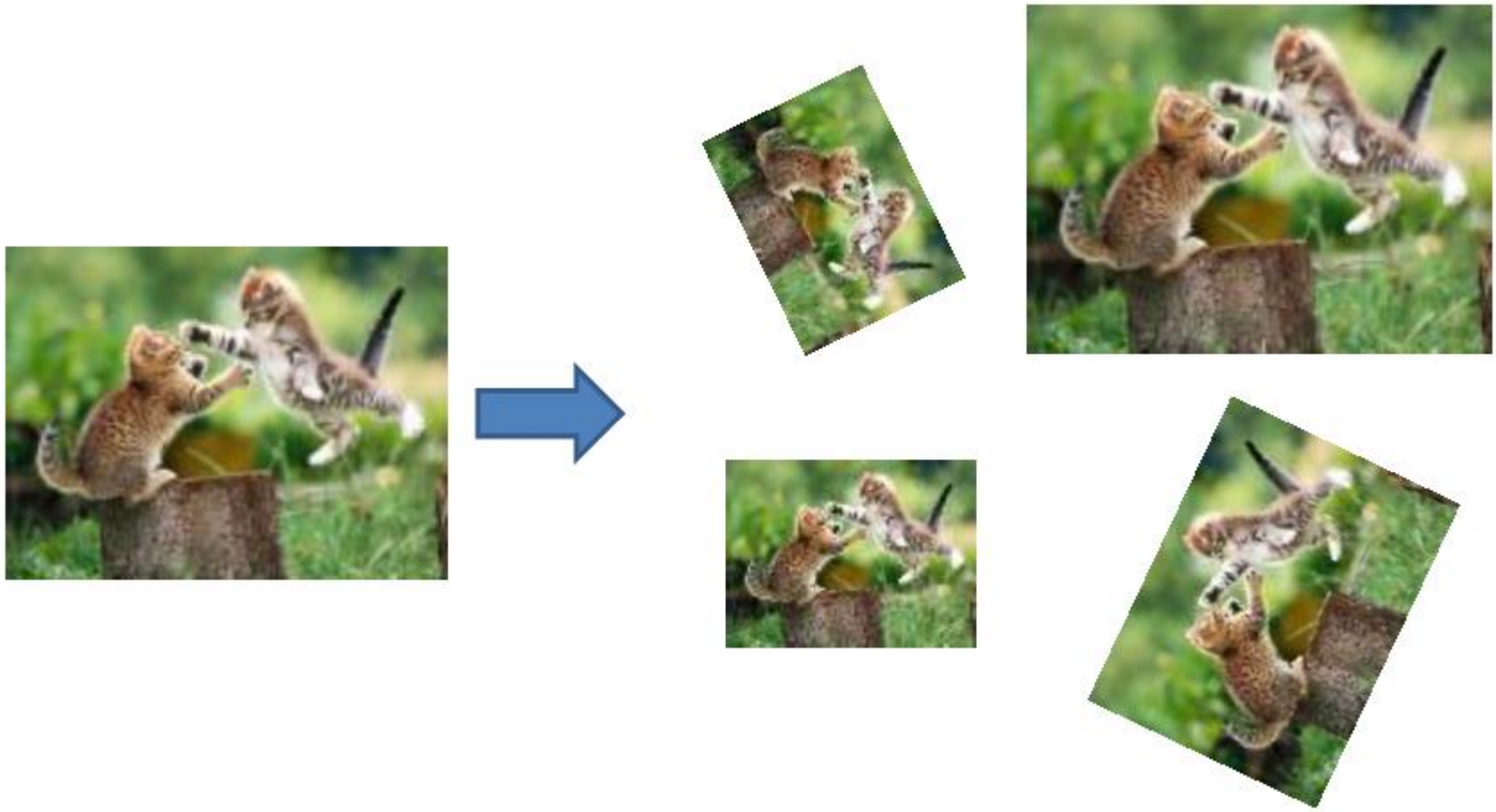
Image Transformation

Sung Soo Hwang

Introduction

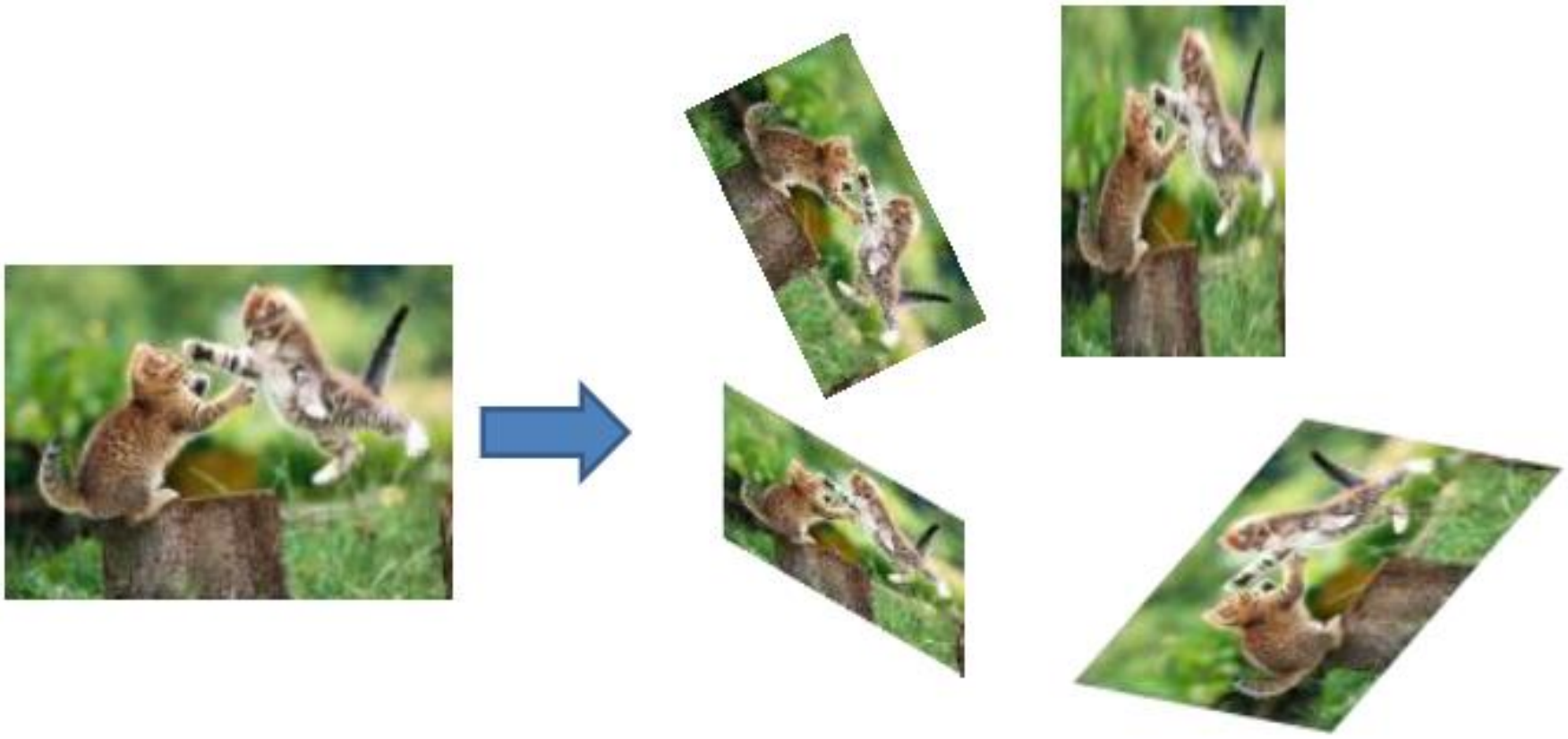
- Similarity transformation
  - Shape of an object is preserved



Images from "Geometry Transformations," by Larry Zitnick, https://courses.cs.washington.edu/courses/csep576/11sp/pdf/Transformations.pdf

Introduction

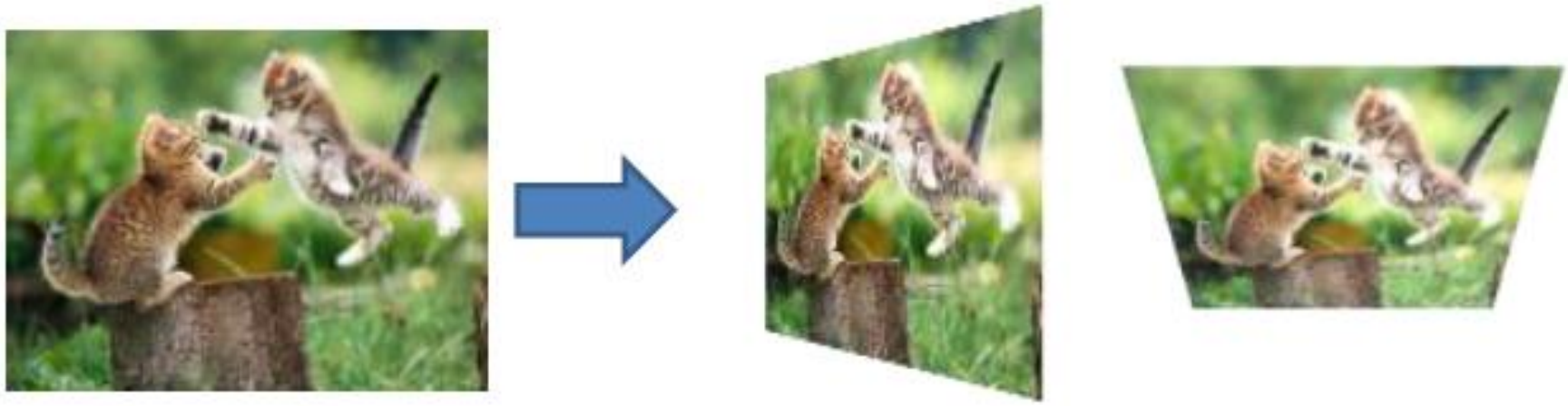
- Affine transformation
  - Parallel lines are parallel after transformation

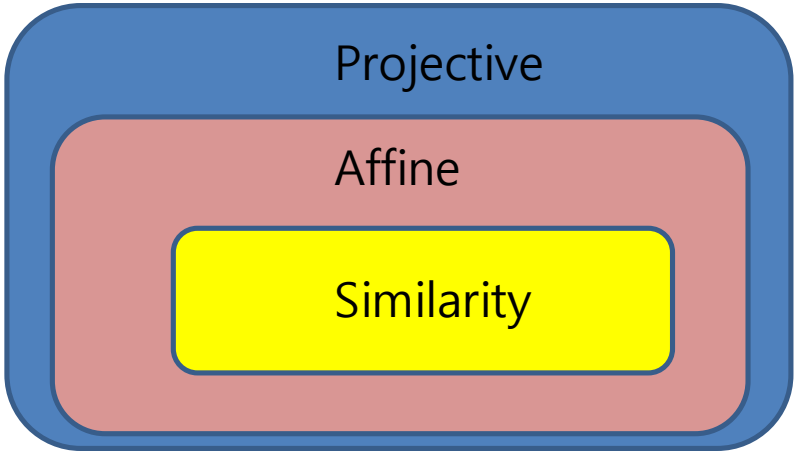


Images from "Geometry Transformations," by Larry Zitnick, https://courses.cs.washington.edu/courses/csep576/11sp/pdf/Transformations.pdf

Introduction

- Projective transformation
  - Lines ad lines after projective transformation

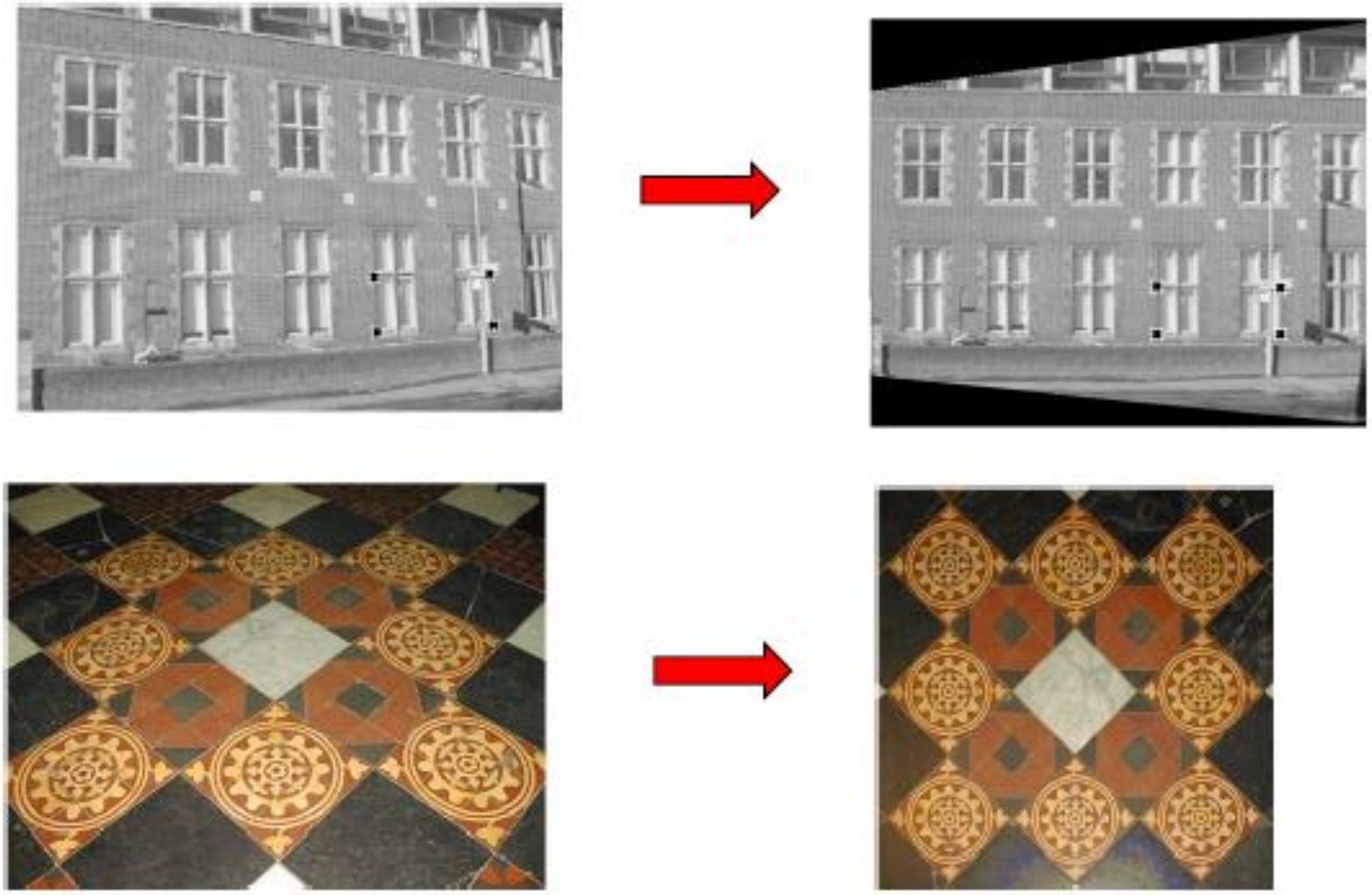




Images from "Geometry Transformations," by Larry Zitnick, https://courses.cs.washington.edu/courses/csep576/11sp/pdf/Transformations.pdf

Introduction

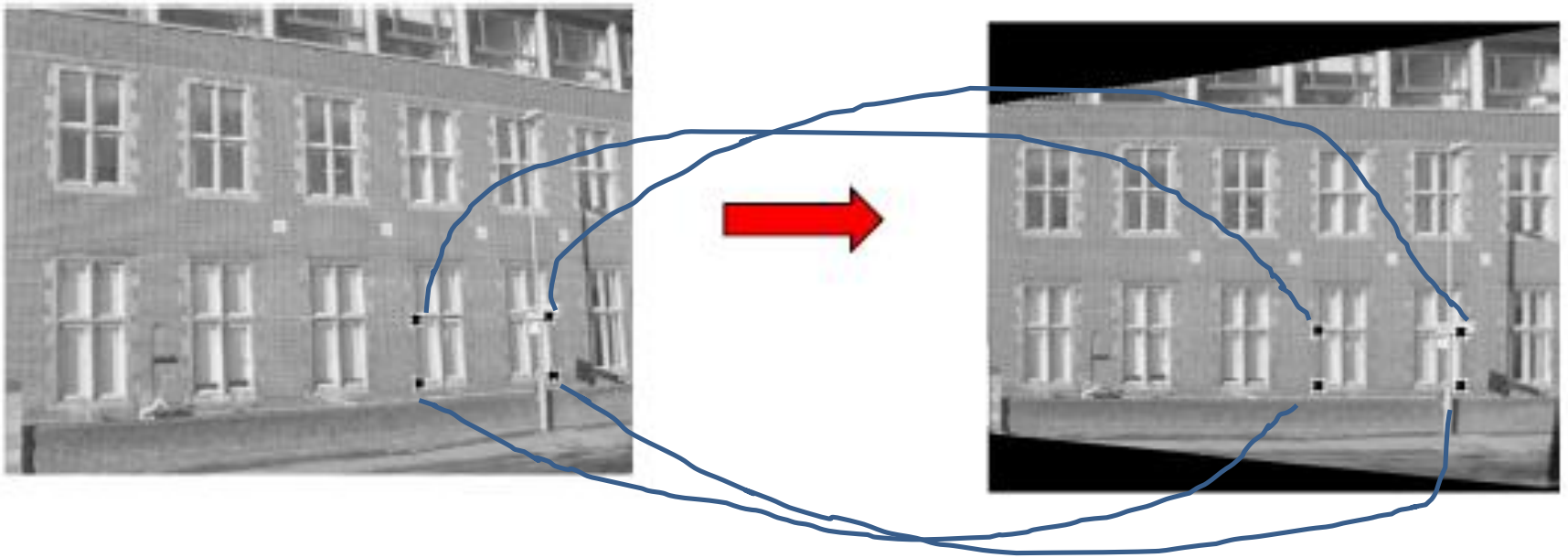
- Projective transformation = Perspective transformation = Homography



Perspective Transformation

- In order to apply perspective transformation, a matrix which explains the relationship between two images should be calculated
  - The dimension of the matrix is 3X3
  - However, only 8 elements in the matrix should be known

➔ At least 4 corresponding pair should be given!



Perspective Transformation

- Example code

```
struct MouseParams
{
   Mat img;
   vector<Point2f> in, out;
};


static void onMouse(int event, int x, int y, int, void* param)
{
   MouseParams* mp = (MouseParams*)param;
   Mat img = mp->img;
   if (event == EVENT_LBUTTONDOWN) // left button
   {
      Mat result;

      //왼쪽 상단부터 시계방향으로 입력
      mp->in.push_back(Point2f(x, y));


      if (mp->in.size() == 4)
      {
         //네 쌍의 매칭쌍으로 부터 homography 행렬계산
         Mat homo_mat = getPerspectiveTransform(mp->in, mp->out);
```

Perspective Transformation

- Example code

```
//homography 행렬을 이용한 warp
        warpPerspective(img, result, homo_mat, Size(200, 200));
        imshow("output", result);
    }
    else
    {
        result = img.clone();
        for (size_t i = 0; i < mp->in.size(); i++)
        {
            circle(result, mp->in[i], 3, Scalar(0, 0, 255), 5);
        }
        imshow("input", result);
    }
}

//클릭 reset
if (event == EVENT_RBUTTONDOWN)
{
    mp->in.clear();
    imshow("input", img);
}
}
```

Perspective Transformation

- Example code

```
int main()
{
    Mat imput = imread("book.PNG");
    imshow("input", imput);

    MouseParams mp;
    mp.out.push_back(Point2f(0, 0));
    mp.out.push_back(Point2f(200, 0));
    mp.out.push_back(Point2f(200, 200));
    mp.out.push_back(Point2f(0, 200));
    mp.img = imput;

    setMouseCallback("input", onMouse, (void*)&mp);
    waitKey();
    return 0;
}
```