Different approaches to Frame Interpolation and Motion Interpolation

Autors:

Pablo Bernal Moreno, pbernal2@eafit.edu.co

Diego Andres Giraldo Gomez, dgiral58@eafit.edu.co

Marco Antonio Franco Montoya, mfranc18@eafit.edu.co

Lorenzo Cannafarina Carrizosa, lcannafa@eafit.edu.co

Francisco José Correa Zabala, fcorrea@eafit.edu.co

System Engineering Department, EAFIT University Medellín

Abstract

This investigation is aimed toward the learning and understanding of different methods that can be used to interpolate certain frames based on previous and future frames in order to grant a greater fluidity to the media, We're trying to apply different interpolations to the colors of each and every pixel to determine a function (using Newton's Interpolation, Natural Cubic Splines and Cubic B-Spline) in which the color of said pixel changes frame by frame allowing us to estimate the color that would allow a smooth transition between frames. While we're trying to do our own method, we are also doing a research about the state of the art methods that also implements Machine Learning techniques such as object recognition and movement detection. Some of those techniques are: DAIN (Depth-Aware Video Frame Interpolation) and ISOMAP interpolation.

Introduction

In this investigation we are going to present different approaches attempting to resolve the problem of constructing frames between already defined frame, also called "Frame Interpolation", the objective is that by creating intermediate frames we can achieve higher fluidity and better visual quality when reproducing videos or GIFs while using higher numbers of Frame Per Seconds. For this purpose we are going to introduce different methods of interpolation and machine learning that interpolate a function and with said function produce a "transition frame". We represent the frame has a matrix of the resolution of said frame alongside a matrix RGB which represents the color of a determined pixel. Another idea that we want to explore is which interpolation method and which machine learning method are the best for this purpose and the method employed to smoothen the videos using transition frames and machine learning.

Using image interpolation for smoother video has multiple benefits, including compression in the streaming era. The amount of content accessible on the internet requires both data and the bandwidth to provide it. Interpolating frames to reconstruct compressed images provides a similar experience and requires less resources.

Objectives

General:

● Find out which interpolation and machine learning method is more appropriate for

frame interpolation depending of the context

Specifics:

● Construct a transition frame of a video using interpolation.

● Investigate methods to smooth the frame transition of a video using machine learning and frame interpolation.

Theoretical Framework

Images representation in Computers

Resolution

In order to store and reproduce images or in our case "Frames" the computer requires a bit by bit format, these are called Pixels, and for the computer an image is a collection of colored pixels defined by the resolution of the image, when an image is defined 800x600 means that the image is composed by 480.000 pixels aligned in a square like form of 600 row (Width) and 800 column (Height)

Images Formats

RGB

The RGB color model is an additive color model[1] in which red, green and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.

When used in computers the image is defined by a matrix of pixels in the format $[width] x\ [height]\ x\ [3]$ where the third parameter are the intensity of Red Blue and Green that the pixel shines

Using the RGB format we can describe an image and the color of every part of it we can compare two frames using the difference between the RGB data of equally situated pixels, places where the color matrix are different represent pixel that are part of the movement

Video

For research purposes the data will be provided by the Davis Challenge 2017 [1] Semi supervised Dataset, which contains a series of videos separated by its frames in separate folders. Each frame is stored in the .jpg format as an image, in 480p resolution.

Interpolation

Interpolation is a method used in this case to allow the construction of new data points within the range of an existing set of data points. In order to use interpolation, data must be simulated or obtained using experimentation, and it is often required to estimate the value of the interpolation function for an intermediate value of the independent variable.

Like any supervised method used in Machine Learning, using a clean dataset is key to avoid multicollinearity. In some cases it is not recommended to use all the available variables. By using Interpolation, we attempt to find a function that mimics the behaviour of the response variable by using the previously established independent variables. In this case we will implement the following interpolation methods: Newton, Lagrange, Linear, Cubic.

To introduce interpolation, is important to introduce the following theorem:

Given n + 1 points $(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ *with the condition that* $x_i \neq x_j$ *for each i,j such that* $0 \leq i \leq n$ *and* $0 \leq j \leq n$, *then there exits a unique polynomial* $p(x)$ *of degree at most n with the property:*

$$p(x_i) = y_i \quad i = 0, 1, 2, \ldots, n$$

Figure 1

[2]

Newton interpolating polynomial:

From previous theorem, is deduced that the Newton interpolation polynomial has the form: [3]

$$p_n(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \ldots + b_n(x - x_0)(x - x_1)\ldots(x - x_n)$$

Figure 2

However, the process to obtain each b term is so long. For this reason we introduce another technique that is known as Newton interpolating polynomial with divided differences. Which is an incremental process in which we obtain an interpolating polynomial in each step [2].

Splines

is a special function defined piecewise by polynomials. In interpolating problems, spline interpolation is often preferred to polynomial interpolation because it yields similar results, even when using low degree polynomials, while avoiding Runge’s Phenomenon for higher degrees. Following the next formula:

$$S : [a, b] \to \mathbb{R}.$$

$$[t_i, t_{i+1}] \, , i = 0, \ldots, k-1$$
$$[a, b] = [t_0, t_1] \cup [t_1, t_2] \cup \cdots \cup [t_{k-2}, t_{k-1}] \cup [t_{k-1}, t_k]$$
$$a = t_0 \leq t_1 \leq \cdots \leq t_{k-1} \leq t_k = b$$

$$P_i : [t_i, t_{i+1}] \to \mathbb{R}.$$

$$\begin{aligned} S(t) &= P_0(t) \, , t_0 \leq t < t_1, \\ S(t) &= P_1(t) \, , t_1 \leq t < t_2, \\ &\vdots \\ S(t) &= P_{k-1}(t) \, , t_{k-1} \leq t \leq t_k. \end{aligned}$$

Figure 3

Natural Cubic Spline interpolation

Using the scipy library CubicSpline we can have results that will interpolate a value using the equations with the form [3] [5]:

$$P(x) = ax^3 + bx^2 + cx + d$$

Figure 4

With four unknown variables for each interval. Then we find more equations using the first and second derivative

Cubic B-Spline interpolation

A cubic spline that uses the minimum knots, the scipy interpolation methods follow the next algorithms [3] [6]:

$$\mathbf{S}_i(t) = \sum_{k=0}^{3} \mathbf{P}_{i-3+k} b_{i-3+k,3}(t) \; ; \; t \in [0,1]$$

Figure 5

$$\mathbf{S}(t) = \sum_{i=0}^{m-1} \mathbf{P}_i b_{i,3}(t)$$

Figure 6

where S*i* is the *i* B-Spline segment and P its the vector of control points.

Both Cubic Spline and Cubic B-Splines are used because they (unlike polynomial interpolation methods) don't have the Runge's Phenomenom

Machine Learning

Machine learning is a field of Artificial Intelligence. It is used to learn from data with diverse approaches, and some of its uses involve the recognition of data (objects from an image, a genre of music, etc.), or the prediction of new data points (potential sales in a company, stock market growth).

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Figure 7

Machine Learning can be divided in two main segments

- **Supervised:** It uses previously known data to recognize the behaviour of a response variable
  - **Predictive:** It uses current data to predict future outcomes
    - Neural Networks
  - **Classification:** Uses current data to recognise its behaviour regarding a response variable
    - Logistic Regressions
    - Random Forest
    - Decision Trees
- **Unsupervised:** This type of model adapts to outcomes, observations without previous knowledge

The interpolation Methods used in this case would be classified as a Supervised Machine Learning Method

Development

Environment

This project was developed using the following tools:

- Python 3.7
  - Numpy
  - Pandas
  - Sympy
  - Scipy
  - PIL
- Anaconda 3
  - Jupyter Notebooks
- The Davis Challenge Database Non Supervised 2017

Code

Source code can be found at the following GitHub repository

The Jupyter Notebook : code

Basic Functions

this code segment implements the required functions to use Newton Interpolation, and both Spline Interpolations.

Format Data

This code is used to get the RGB values of various images in jpg format into dataframes

Getting Test Images

in this case we get the images from the Davis Dataset

Data interpolation

using the dataset we selected four images from a video and got the numeric values for the colors of each pixel. In this case we will implement multiple interpolation methods between image1 and image4, these method are Newton's Differentiation, Natural Cubic Spline and Cubic B-Spline

Output

the result of these algorithms can be shown as a DataFrame of each RGB parameter for each pixel, or the resulting image of the interpolation process

	r	g	b
0	43.8125	38.8125	35.7500
1	41.6875	36.6875	33.6250
2	39.1875	34.1250	30.9375
3	46.8125	41.5625	38.3750
4	49.0000	44.8125	41.5625

Figure 8

	r	g	b
0	43.2500	38.25000000000001	35.1875
1	41.1250	36.125	33.0625
2	38.6250	33.5625	30.375
3	46.2500	41.0	37.8125
4	48.4375	44.25	41.00000000000001

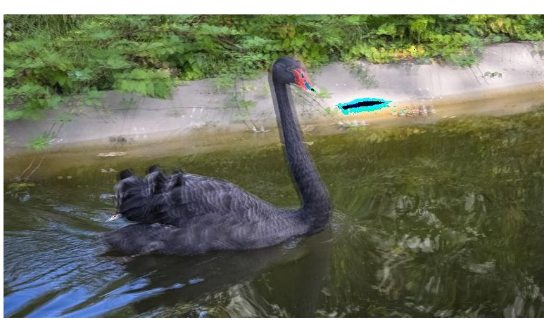
Figure 9

	r	g	b
0	43.100	38.199999999999996	35.225
1	40.750	35.85	32.875
2	38.150	33.175000000000004	30.049999999999997
3	46.000	40.8	37.675000000000004
4	48.225	44.1	40.9

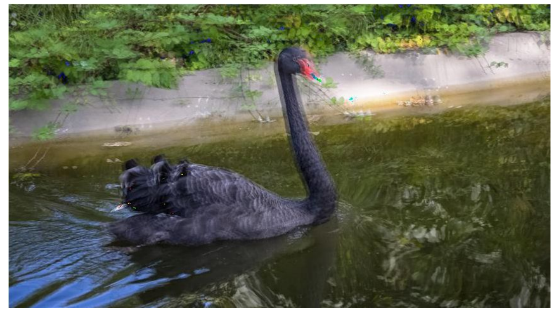
Figure 10

Images

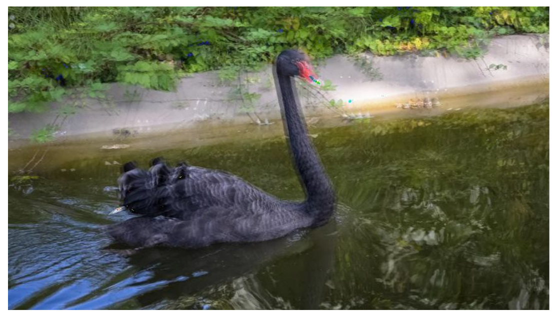
Newton



B Cubic Splines



Cubic Splines



Error Analysis

the error of these methods was calculated by analysing separately the rgb parameters of the interpolated image with the real frame from the video in the dataset.

$$\frac{mean(abs(interpolatedimage_{rgb} - interpolatedimage_{rgb}))}{255}$$

the results for each method were:

Using Cubic Splines

Comparing Results with Reality

```
In [70]: df_comparison = (df_interpolation - df_img1).abs()
         df_comparison.mean()
         df_test = ( df_comparison.mean() / 255 ) * 100
         df_test

Out[70]: r    3.966371
         g    4.144197
         b    4.121521
```

Using B Cubic Splines

Comparing Results with Reality

```
df_comparison = (df_interpolation - df_img1).abs()
df_comparison.mean()
df_Bspline = ( df_comparison.mean() / 255 ) * 100
df_Bspline
```

```
r    4.965853
g    5.184416
b    5.136665
```

Using Newton

Comparing Results with Reality

```
df_comparison = (df_interpolation - df_img1).abs()
df_comparison.mean()
df_newton = ( df_comparison.mean() / 255 ) * 100
df_newton
```

```
r    4.987289
g    5.208736
b    5.157071
```

State Of The Art

Interpolating images between video frames using non-linear dimensionality reduction

Video interpolation implies the three-dimensional movement of an object in a two-dimensional field, and in some instances simple image interpolation is not enough. For those instances a potential solution is using low dimensional methods such as Isomap that allow the recognition of the objects in order to transform the movement of the object in a three-dimensional space. [7]

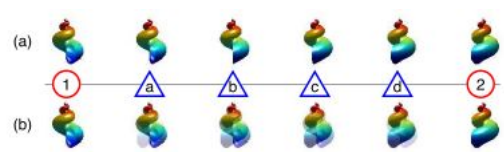


Figure 11

Depth-Aware Video Frame Interpolation

also called the DAIN is a state of the art method for motion and frame interpolation proposed by Shanghai Jiao Tong University in collaboration with google, University of California, Merced y licensed por el MIT

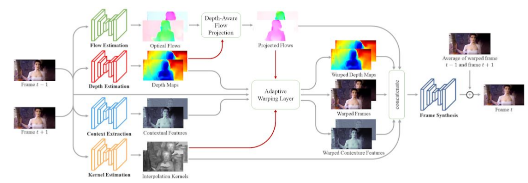


Figure 12

Depths is the main focus and innovation of this method, by using the Depth and Projected flow the method can ascertain which objects are closer or away from the focal point, “the projected flows tend to sample the closer objects and reduce the contribution of occluded pixels which have larger depth values”, the main problem with the depth flow projection is that sometimes it might exist that none of the vector aims toward a certain position in the flow for those cases the average of the neighboring position is used to compute in order to fill those gaps [7]

Conclusions

While analysing the resulting images of the interpolated and original frames, it is apparent the generation of motion blur in the frames, and the superiority of the results generated using cubic splines and b cubic splines compared to Newton interpolation.

This result can be seen especially in the edges of moving objects in the frame given Runge's Phenomenon.

Comparing the results of B Cubic Splines and Cubic splines, we can also conclude that B Cubic Splines has the advantage of using the minimum quantity of nodes, which results in the minimum polynomial order, giving better results even when the distance between frames increases.

During our testing it was also shown the variation of the results when using frames with increasing distance between them. The less distance between frames, the better

It is also shown that in some scenarios there is a tradeoff between motion blur and pixel precision comparing some algorithms. The decision of using any of these interpolation methods would rely on the use case.

However, the implementation of these interpolation methods implies high algorithmic complexity because of the execution of these methods for each color of each pixel in the image. In order to scale the use of these algorithms it is required the use of object detection algorithms that narrow the data and optimize performance.

Future Work

Movement Detection

in order to preserve image fidelity recognizing the edges of an object and its center of mass using scipy could provide smoother frame interpolation, given that the movement would be analyzed in the context of the moving object.

Figure 13

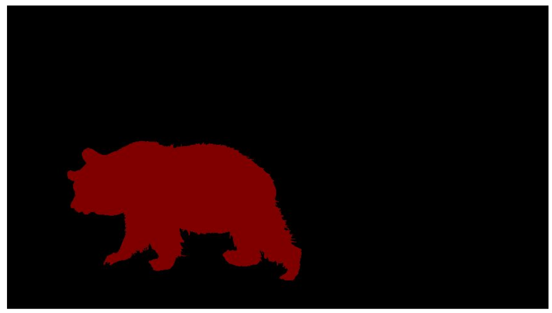


Figure 14

Object Detection

Object detection in video provides a way to compress repetitive data such as backgrounds. For this application, only a few frames of the background are preserved while others are simply expressed as the previous background.

This approach can provide smoother movement frames, given that only the moving object images would be changed, reducing the potential error of processing every pixel of the image.

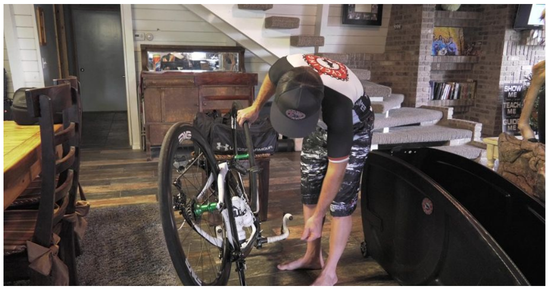


Figure 15

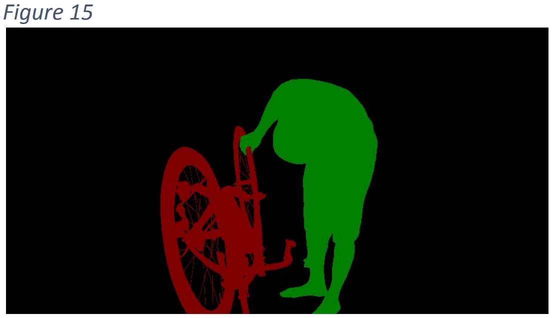


Figure 16

Attachments

Figure 1 3

Given $n+1$ *points* $(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ *with the condition that* $x_i \neq x_j$ *for each* i, j *such that* $0 \le i \le n$ *and* $0 \le j \le n$, *then there exits a unique polynomial* $p(x)$ *of degree at most* n *with the property*:

$$p(x_i) = y_i \quad i = 0, 1, 2, \ldots, n$$

Figure 2 3

$$p_n(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \ldots + b_n(x - x_0)(x - x_1)\ldots(x - x_n)$$

Figure 3 3

$$P(x) = ax^3 + bx^2 + cx + d$$

Figure 4 4

$$\mathbf{S}_i(t) = \sum_{k=0}^{3} \mathbf{P}_{i-3+k} b_{i-3+k,3}(t)\ ;\ t \in [0,1]$$

Figure 5 4

$$\mathbf{S}(t) = \sum_{i=0}^{m-1} \mathbf{P}_i b_{i,3}(t)$$

Figure 6 4

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Figure 7 5

	r	g	b
0	43.8125	38.8125	35.7500
1	41.6875	36.6875	33.6250
2	39.1875	34.1250	30.9375
3	46.8125	41.5625	38.3750
4	49.0000	44.8125	41.5625

Figure 8 5

	r	g	b
0	43.2500	38.25000000000001	35.1875
1	41.1250	36.125	33.0625
2	38.6250	33.5625	30.375
3	46.2500	41.0	37.8125
4	48.4375	44.25	41.00000000000001

Figure 9 5

	r	g	b
0	43.100	38.199999999999996	35.225
1	40.750	35.85	32.875
2	38.150	33.175000000000004	30.049999999999997
3	46.000	40.8	37.675000000000004
4	48.225	44.1	40.9

Figure 10 6

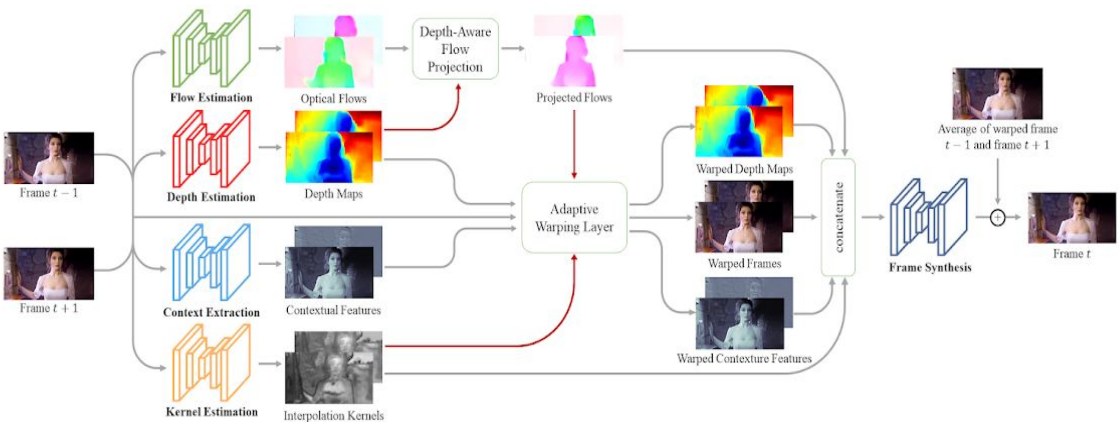


Figure 11 6

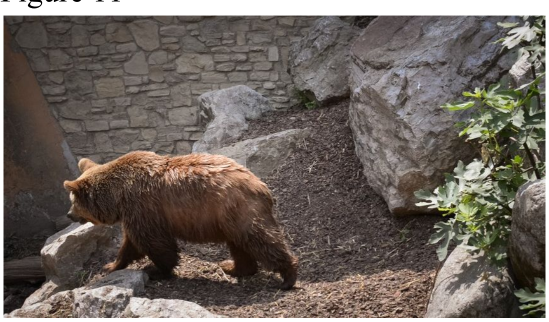


Figure 12 7

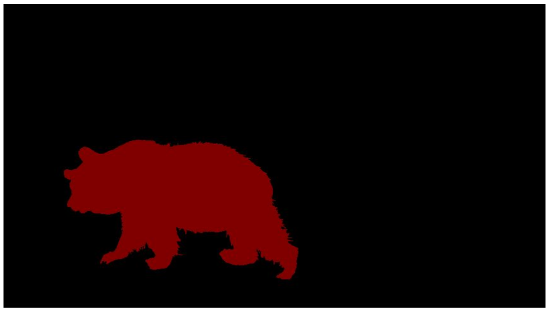


Figure 13 7

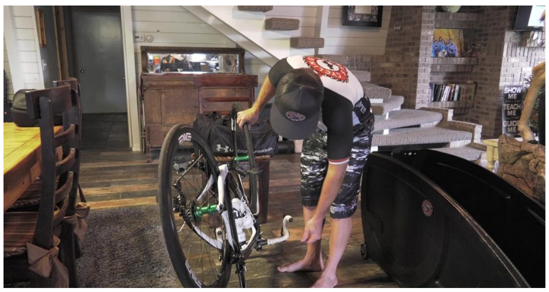
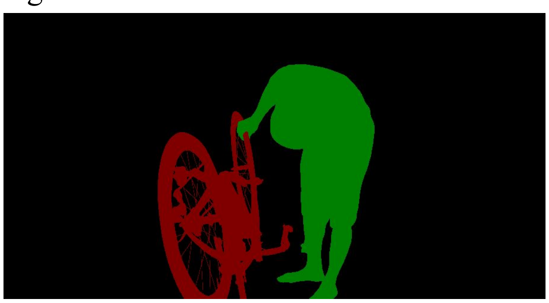


Figure 14 7



References

[1] "Davis Challenge," 2017. [Online]. Available: https://davischallenge.org/davis2017/code.html.

[2] Scipy, "Scipy Documentation," [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.splrep.html.

[3] F. J. C. Zabala, Métodos Numéricos, Fondo Editorial Universidad EAFIT, 2010.

[4] R. Hirsch, Exploring Colour Photography: A Complete Guide., Laurence King Publishing. ISBN 1-85669-420-8., 2004.

[5] Scipy, "Scipy Documentation Splines," [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.splev.html.

[6] Scipy, "Scipy Documentation Cubic Splines," [Online]. Available: https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.interpolate.CubicSpline.html.

[7] S. R. and S. EL GHAZZAL , "Interpolating images between video frames using non-linear dimensionality reduction," [Online]. Available: http://cs229.stanford.edu/proj2012/ElGhazzalRobaszkiewicz-InterpolatingImagesBetweenVideoFramesUsingNonLinearDimensionalityReduction.pdf.

[8] W. o Bao, L. Wei-Sheng, . Z. Xiaoyun, G. Zhiyong and Y. Ming-Hsuan , "Depth-Aware Video Frame Interpolation," 2019. [Online]. Available: https://arxiv.org/pdf/1904.00830v1.pdf.