HyperLapse

Shashank Yadav
Pranjal Maheshwari

Introduction

Hyperlapse (also walklapse, spacelapse, stop-motion time-lapse, motion timelapse, moving timelapse) is an exposure technique in time-lapse photography, in which the position of the camera is being changed between each exposure in order to create a tracking shot in timelapse sequences.

In short :

Time Lapse with motion !!

Example

Problem Statement

Given a video and a speedup rate (2X, 3X,4X or even 10X) create a hyperlapse of a given video.

Sounds easy but it is not.

Naive Solution

To get a speedup x sample 1 frame after every x.

Issue:

Bad transition between frames because of motion, leads to shaky video.

Other Solutions

1. Hardware based : Instagram app on iPhone , uses gyro and speedometer for stabilization
2. Software based:
   (a) Sequentially Perform stabilization and generate timelapse
   (b) SFM for 3D-reconstruction and then stabilization
   (c) Stabilization and Timelapse together : Our approach

Outline of solution

1. **Frame-matching :** using sparse feature-based techniques we estimate how well each frame can be aligned to its temporal neighbors.
2. **Frame selection:** a dynamic-time-warping (DTW) algorithm to find an optimal path of frames that trades-off matching the target rate with minimizing frame-to-frame motion.
3. **Path smoothing and rendering:** given the selected frames, smoothing the camera path to produce a stabilized result.

Cost Function

1. **Frame Matching Cost:**

   To account for motion between the frames. Helps in generating smooth transition between frames.

2. **Velocity and acceleration Cost :**

   To account for deviation from the target speedup.

Frame Matching Cost

$$C_r(i,j)=\frac{1}{n}\sum_{p=1}^{n}\left\|(x_p,y_p)_j^T - T(i,j)(x_p,y_p)_i^T\right\|_2$$

$$C_o(i,j)=\left\|(x_0,y_0)^T - T(i,j)(x_0,y_0)^T\right\|_2$$

$$C_m(i,j)=\begin{cases} C_o(i,j) & C_r(i,j)<\tau_c \\ \gamma & C_r(i,j)\geq\tau_c \end{cases}$$

Velocity and acceleration cost

$$C_s(i,j,\nu) = min(||(j-i)-\nu||_2^2, \tau_s)$$

$$C_a(h,i,j) = min(||(j-i)-(i-h)||_2^2, \tau_a)$$

Optimal Frame Selection

$$C(h,i,j,\nu) = C_m(i,j) + \lambda_s C_s(i,j,\nu) + \lambda_a C_a(h,i,j)$$

$$\phi(p,\nu) = \sum_{\tilde{t}=1}^{\tilde{T}-1} C(p(\tilde{t}-1), p(\tilde{t}), p(\tilde{t}+1), \nu)$$

$$\mathbf{p}_\nu = \operatorname*{argmin}_p \phi(p,\nu)$$

Algorithm

1. **Stage 1:**
    a. The matching cost is computed using frame matching and is stored in a sparse, static cost matrix Cm for all frames <1,2,...,T>.
    b. As an optimization we compute a banded (or windowed) version of C
    c. For a particular input video and value of w, Cm is static and computed once and re-used for generating any speed up v <=w

Algorithm Contd.

1. **Stage 2:**
   a. Stage 2 is the DP algorithm and it consists of two passes.
   b. A first pass populates a dynamic cost matrix Dv.
   c. Dv(i, j) represents the running minimal cost path that ends with the frames i and j.
   d. Once D is fully populated the second pass finds an optimal path by finding the minimal cost in the final rows and columns of Dv

Path Smoothing and rendering

1. Compute a smooth camera motion path and warp the images to generate the result.
2. This step is essentially the same as running standard video stabilization.

Results

Naive Approach

Results

Using Optimal Frame Selection

Thanks!

Code Available at :
https://github.com/shashank-yadav/hyperLapse

Based on :
N. Joshi et. al, Real-Time Hyperlapse Creation via Optimal Frame Selection