

Freeze Me! - Dokumentation

A Media Processing Project

Stand: 28.02.2023

Gruppe: Jan Rekemeyer, Iskander Yusupov und Hendrik Finke

1 Motivation

Da das Ziel des Projekts darin bestand, ein Softwareprodukt zu erstellen, entschied sich die Gruppe, eine Motivation im Marketingformat zu erstellen.

Glückliche Momente lassen sich in einem Foto festhalten, das dann verschenkt werden kann. Aber was tun mit den glücklichen Momenten, die auf Video festgehalten wurden? Mit diesem Ansatz suchte die Gruppe nach Möglichkeiten, ein Video so in Form eines Bildes festzuhalten, dass dieses später beispielsweise als gerahmtes Geschenk präsentiert werden könnte. Eine weitere Inspiration für unsere Gruppe waren Videos mit viel Dynamik. Sport, Tanz, aber auch Tiktok-Challenges enthalten viel Bewegung. Beim Fotografieren von sich bewegenden Objekten erzeugt die Einstellung einer Langzeitbeleuchtung den Effekt einer „eingefrorenen“ Bewegung. Diese Fotos haben ihre eigene Ästhetik und unsere Gruppe wollte versuchen, diesen Effekt mit den im Modul erlernten Fähigkeiten zu reproduzieren.

2 Zielsetzung

Unser Projekt haben wir der folgende Zielsetzung gewidmet:

Wir entwickeln ein Programm, welches es ermöglicht aus einem Video ein Bild zu erzeugen, welches die Bewegungsdynamik des Videos ästhetisch einfasst und dabei den Eindruck fortwährender Bewegung erhält.

3 Langzeitbelichtung - Inspirationsquelle

3.1 Motivation

Das in den Eröffnungsfolien gezeigte Bildmaterial erinnerte uns in Teilen doch recht stark an eine klassische Langzeitbelichtung, wie man sie aus der Analogfotografie oder der Fotografie mit DSLRs oder DSLMs kennt. Daher stellten wir uns die Frage, wie man dieses Konzept digital nachträglich unter Zuhilfenahme eines Videos anstelle eines Fotos reproduzieren kann.

3.2 Funktionsweise

Der Ansatz, den wir hierbei verfolgt haben ist relativ trivial. Ähnlich wie klassische Filter den Inhaltswert eines Pixels anhand einer ggf. faktorisierten Durchschnittsberechnung mit den umliegenden Pixeln berechnet tun wir ebendieses - jedoch mit den Pixeln gleicher Koordinaten auf anderen Videoframes. Zu diesem Zwecke werden Videos Frame für Frame betrachtet und dann Pixel für Pixel in ihre jeweiligen 3 Farbkomponenten zerlegt. Im folgenden wird der Durchschnittswert der Farbwerte errechnet und somit eine Art Langzeitpixel erzeugt, deren Summe das künstlich Langzeitbelichtete Bild ergibt.

3.3 Resultate

Die Resultate waren abhängig vom eingegebenen Video durchaus vielversprechend. Ein von uns anfänglich eigens erstelltes Video von der Kreuzung der Ammerländer Heerstraße mit dem Uhlhornsweg resultierte in einem mehr oder minder matschigen Bild. Das Ergebnis des Contemporary Dancers hingegen war schon erstaunlich dicht an den Hintergrundscheinungen der Fotos in den Veranstaltungsfolien. Auch das TikTok Video mit dem Sprung resultierte in einem relativ ansehnlichen Ergebnis. Da dieser Ansatz somit der erste nennenswerte Schritt in die Richtung des Gesamtzieles war wurde er beibehalten und die Langzeitbelichtung wurde somit Grundkomponente unseres Projektes.

4 Edge Detection und Background Subtraction

4.1 Motivation

Nachdem es der Gruppe gelungen war, Bilder mit einer künstlichen Langzeitbelichtung aus dem Video zu erstellen, fiel ein Merkmal auf. Wurde das Video ohne Stativ oder jegliche Art von Unterstützung gedreht, dann führte das leichte Wackeln der Kamera irgendwann dazu, dass sich auch stabile Objekte im Ausgabebild als unscharf herausstellten. Um dieses Merkmal oder Problem zu lösen, wurde entschieden, Canny Edge Detection zu verwenden. Die Idee, der Algorithmus zu verwenden, wurde experimentell geboren. Wir haben versucht, Canny Edge Detection auf ein bereits bearbeitetes Bild anzuwenden. Die Methode erkannte nur Objekte als Kanten, die auf dem Video stabil waren. Dann entschied die Gruppe, dass es möglich ist, Schärfungsverfahren (z. B. einen Laplace-Filter) auf die Pixelkoordinaten mit stabilen Objekten anzuwenden, die mithilfe des Canny Edge Detection auf dem Ausgabebild erhalten werden.

4.2 Theorie

Der Canny Edge Detection ist ein mehrstufiger Algorithmus zum Erkennen von Kanten in einem Bild. Der Algorithmus besteht aus vier Stufen. In der ersten Stufe ist es notwendig, Rauschen aus dem Originalbild zu entfernen, für diese Aufgabe wird der Gauß Filter angewendet. Der zweite Stufe besteht darin, den Größte entlang der x- und y-Dimension zu erhalten. Dazu wird die Ableitung des Gauß-Filters berechnet, um dann das Gradient

der Bildpixel zu berechnen. Der Algorithmus geht alle Punkte der Gradientenintensitätsmatrix durch und findet die Pixel mit dem maximalen Wert in den Kantenrichtungen. Das Betrachten der Gruppe von Nachbarn für jede Kurve in einer Richtung senkrecht zu der gegebenen Kante unterdrückt nicht maximalen Kantenbeitragspixelpunkte. Schließlich wird das Hysterese-Schwellenwertverfahren verwendet, um die Pixel zu bewahren, die höher als die Gradientengröße sind, und diejenigen zu vernachlässigen, die niedriger als der niedrige Schwellenwert sind.

4.3 Codebeschreibung

Zur Optimierung der Ergebnisse wird ein Verfahren zur Bestimmung der Schwellwerte (minVal und maxVal) eingesetzt. Die Schwellwerte werden dann in der OpenCV Methode `cv.Canny` übergeben.

4.4 Ergebnis

Nach dem Ausführen des Codes ist das erwartete Ergebnis ein Bild mit identifizierten Kanten. Nach Auswertung der Ergebnisse entschied die Gruppe, dass der resultierende Bildschärfungseffekt nicht effektiv genug war, und entschied sich daher, Canny Edge Detection aufzugeben.

5 Edge Detection und Background Subtraction

6 Fazit

Das letztendliche Ergebnis unserer Arbeit entspricht unserer Meinung nach dem ursprünglichen Ziel - aufgrund der Subjektivität der Ästhetik aber lässt sich dieses Urteil wohl kaum objektiv fällen. Mit der Anwendung der Langzeitbelichtung fand sich für uns relativ früh eine Möglichkeit, einen Hintergrund mit den verwaschenen bzw. verschwommenen Hintergrundartefakten zu erzeugen.

6.1 Limitierungen

Die Erkennung des Vordergrundes beschränkt die Funktionsweise der Software in Teilen. Von den anfänglich 3 angenommenen Testvideos (Tänzer, Kreuzung, TikTok Sprungvideo) funktioniert lediglich der Tänzer wirklich gut. Das Ergebnis des TikTok Sprungvideos ist ebenfalls durchaus annehmbar, wobei hier das Hinzufügen eines maskierten Standbildes weitestgehend wertlos ist. Zu diesem Zwecke müsste ein Framepicker in der Software implementiert werden. Gleichmaßen ist der Kontrast der Hintergrundeffekte relativ stark beschränkt, was ein Betrachten der Resultate in schlechten Lichtverhältnissen oder gar

auf Beamern stark einschränkt. Um diesen Effekt zu verstärken hätte eine helligkeitsbedingte Faktorisierung genutzt werden können, die wir anfänglich mal betrachtet haben, deren Notwendigkeit uns im Laufe des Projektes nicht mehr sinnvoll erschien. Retrospektiv wäre diese Funktion ein sinniges Opt-In gewesen.

6.2 Ausbaumöglichkeiten

Für eine künftige Ausgestaltung bieten sich die im letzten Block genannten Optionen an: Eine Optimierung des Interfaces um ein Frame zu selektieren, die Option im entsprechenden Frame ein Rechteck zu wählen, das dann an GrabCut übergeben wird um den Vordergrund zu bestimmen. Dazu wäre es eine sicherlich gute Option gewesen die Lichtstärke in die Langzeitbelichtung einfließen zu lassen, um den Kontrast der Schlieren im Hintergrund zu steigern.