

Ostbayerische Technische Hochschule Regensburg
Fakultät für Informatik und Mathematik

Wissenschaftliche Abhandlung

Möglichkeiten zur Bestimmung der visuellen Ähnlichkeit zwischen zwei Bildern

Vorgelegt von: Andreas Huber <andreas.huber@st.oth-regensburg.de>

Matrikelnummer: 3370380

Studiengang: Master Informatik (Schwp. Software Engineering)

Betreuer: Prof. Dr. techn., Dipl.-Ing. Markus Kucera

Abgabedatum: 2. Dezember 2022

Inhaltsverzeichnis

Abbildungsverzeichnis	II
Tabellenverzeichnis	III
Abkürzungsverzeichnis	IV
1 Einleitung	1
2 Algorithmen zur Ähnlichkeitsbestimmung zweier Bilder	1
2.1 Kategorisch durch Schlüsselwörter	1
2.2 Inhaltsbasierte Algorithmen	1
2.2.1 Perceptual Hashing	2
2.2.2 Histogram Intersection	3
2.2.3 Scale-Invariant Feature Transform (SIFT)	4
2.2.4 Wavelet Transformation	4
2.2.5 DeepRanking	4
3 Fazit und Ausblick	5
Quellenverzeichnis	6
Anhang	6

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

1 Einleitung

2 Algorithmen zur Ähnlichkeitsbestimmung zweier Bilder

2.1 Kategorisch durch Schlüsselwörter

2.2 Inhaltsbasierte Algorithmen

In den folgenden Unterkapiteln stellt Bild A jeweils das Suchbild und Bild B das Referenzbild dar.

2.2.1 Perceptual Hashing

Eine der einfachsten, aber auch anfälligsten Methoden zur Bestimmung der Ähnlichkeit von zwei Bildern ist das Hashing. Dabei gibt es viele verschiedene Ansätze und Vorgehensmodelle. Die gängigste Methode ist die Anwendung des pHashes, auch genannt Perceptual Hashing.

Beim Perceptual Hashing werden sowohl für Bild A als auch für Bild B ein Hash berechnet. Die daraus resultierende Hamming-Distanz zwischen den Hashes ergibt die Ähnlichkeit der Bilder. Je geringer die Distanz, desto ähnlicher. Das Verfahren ist nicht normiert und Stand 2. Dezember 2022 noch ein offenes Forschungsthema.

Im Folgenden wird ein beispielhaftes pHash-Verfahren erläutert. Zuerst wird sowohl das Referenz- als auch das Suchbild in eine Graustufen-Grafik umgewandelt. Schließlich wird die Grafik auf 32x32 Pixel skaliert. Auf das entstandene Grauwertbild folgen zwei Diskrete Kosinus Transformationen (1. Pro Zeile, 2. Pro Spalte). Die hochfrequenten Abschnitte befinden sich nun links oben in einer 8x8 Matrix. Daraufhin wird der Median-Grauwert der 64 Pixel berechnet. Jeder Pixel, dessen Grauwert unter dem Durchschnitt liegt wird weiß eingefärbt, der Rest schwarz. Daraus ergibt sich ein 64-Bit langer Hashwert (schwarz: 0, weiß: 1). Zuletzt wird durch eine Subtraktion der beiden generierten Bitfolgen die Hamming-Distanz berechnet. Mittels einem vorher festgelegten Schwellenwert wird jetzt auf Basis des Hamming-Abstands bestimmt, ob die Bilder als ähnlich eingestuft werden.

Vorteile

- Schnelle Performance, wenn die Hashes der Referenzbilder bereits in der Datenbank liegen
- Einfach zu implementieren
- Keine Trainingsdaten nötig
- Robust gegen Wasserzeichen, Farbfilter, leichte Helligkeits- und Kontraständerungen, Gammakorrekturen, Skalierungen sowie Komprimierungen

Nachteile

- Nicht robust gegen Spiegelungen, Rotierungen und Verzerrungen
- Nicht robust gegen Zuschneidungen, neu eingefügten Elementen oder Änderungen des Blickwinkels

2.2.2 Histogram Intersection

Ein robusteres, aber auch rechenaufwendigeres Verfahren ist der Histogram Intersection Algorithmus von Swain und Ballard. Histogramme sind stabil gegenüber Translation und Rotation um die Betrachtungsachse. Außerdem ändern sie sich bei Änderungen des Blickwinkels, des Maßstabs und bei Verdeckung von Elementen nur langsam. Gegen Belichtungsänderungen können stark reduzierte Farbhistogramme jedoch Probleme bereiten.

Im Standardfall wird die Histogram Intersection auf Farbhistogramme angewandt. Zuerst muss die Menge an zu diskretisierenden Farben im Histogramm festgelegt werden - zum Beispiel 100. Folglich gibt es 100 mögliche „Farbeimer“ in die wir jeden Pixel der Bilder einsortieren. Mit den durch das Referenz- und das Suchbild berechneten Histogrammen, kann schließlich die Überschneidung festgestellt werden. Je stärker sich die Histogramme überschneiden, desto ähnlicher sind die Bilder. Zusätzlich wird wie beim Hashing ein gewisser Schwellenwert vorher definiert. Der Vorgang kann beispielsweise auch auf die einzelnen Farbkanäle aufgeteilt werden. Bei RGB würde das bedeuten, dass man jeweils den Rot-, Grün-, und Blau-Kanal einzeln betrachtet.

Weitere Untersuchungen haben ergeben, dass sowohl die Wahl des Farbraums als auch die Festlegung des Quantization Levels (Anzahl der Farbeimer) eine große Rolle bei der Erfolgsschance dieses Vorgehensmodells spielen. Hierbei sorgte wohl der CIELab-Farbraum im Allgemeinen für die besten Ergebnisse.

Eine zusätzliche Möglichkeit die Genauigkeit des Algorithmus zu verbessern ist das Hinzufügen eines sogenannten „Texture Direction“-Histogramms. In diesem Fall wird durch eine Eckenerkennung die Struktur und Richtungen des Bildes bestimmt. Diese werden wie bei den Farbhistogrammen auch in einem Histogramm mit vorher definierten Bins einsortiert. Abschließend kann, wie oben, beschrieben die Überschneidung und somit die Ähnlichkeit der Struktur zweier Bilder mithilfe des „Histogram Intersection“-Algorithmus von Swain und Ballard berechnet werden. Offensichtlich wird jedoch jeder weitere zusätzlicher Berechnungsschritt die allgemeine Performance des Systems beeinträchtigen.

2.2.3 Scale-Invariant Feature Transform (SIFT)

David G. Lowe hat mit der Einführung des SIFT-Algorithmus einen für die Bildverarbeitung wertvollen Beitrag geleistet. Der SIFT-Algorithmus findet lokale Merkmale in Bildern. Diese Merkmale sind invariant gegenüber Bildskalierung und -drehung und teilweise invariant gegenüber Änderungen der Beleuchtung sowie des Betrachtungswinkels. Die Wahrscheinlichkeit einer Störung durch Verdeckung einzelner Elemente oder auftretenden Bildrauschens wird, durch räumliche Streuung und einem vielseitigen Frequenzbereich, erreicht. Im Folgenden wird die grobe Vorgehensweise zur Findung der lokalen Merkmale erläutert.

1. Scale-space extrema detection

Zu Beginn wird aus einem Bild durch repetitives Unschärfen (mittels Gaußschen Unschärfefilter) und Halbieren der Größe eine Serie an Bildern erzeugt. Auf die Bilder der Bilderserie folgt die Anwendung der Gaußschen Differenzfunktion.

2. Keypoint localization

Aus den vorher differenzierten Grafiken werden jetzt mit dem Marr-Hildreth-Operator interessante Schlüsselpunkte bzw. Merkmale herausgefiltert. Eine dem Harris-Corner Detector ähnliche Prozedur wird angewandt, um Punkte auf Linien zwischen Ecken und Punkte mit wenig Kontrast (uninteressante Punkte) herauszusieben.

3. Orientation assignment

Mithilfe eines Histogramms kann den Schlüsselpunkten nun eine Orientierung zugewiesen werden. Dadurch werden die Punkte neben der Robustheit gegenüber Skalierung und Translation auch gegen Rotation invariant.

4. Keypoint descriptor

Zuletzt wird nach Anwendung einiger mathematischer Verfahren eine Art eindeutiger Fingerabdruck des Merkmals berechnet. Schließlich wird dadurch die Stabilität gegenüber begrenzter lokaler Formverzerrungen und Beleuchtungsänderungen erreicht.

Nach Anwendung dieses Algorithmus muss man bei einem 500x500 Pixel großen Bild mit in etwa 2000 robusten Merkmalen rechnen. Werden die gefundenen Merkmale in einer Datenbank gespeichert, können sie schließlich in nahezu Echtzeit auf die Schlüsselpunkte eines Suchbilds verglichen werden.

2.2.4 Wavelet Transformation

2.2.5 DeepRanking

3 Fazit und Ausblick

Anhang