

Wo ist Walter

Suche mit Hilfe von Bild-Operatoren in Matlab

MARCO FÜLLEMANN
JANOSCH ROHDEWALD

Zusammenfassung

In dieser Arbeit wurde mittels Computer-Perception-Algorithmen versucht Walter in einem "Wo ist Walter"-Bild zu finden. Zu diesem Zweck wurden Pullover/Mütze und Brille extrahiert um mögliche Positionen zu finden. Anschliessend wurden die Resultate beider Extraktionen kombiniert und den kleinsten Abstand zwischen potenziellen Pullovern/Mützen und Brillen eruiert. Beim kleinsten Abstand wird Walter vermutet.

Einleitung

Um Walter zu erkennen wurden in zwei unabhängigen Schritten sein Pullover/Mütze und seine Brille extrahiert. Bei der Brillenextraktion wurde eine Kreis-Hough-Transformation mit bestimmten Radien angewandt. Bei der Extraktion seines Pullovers/Mütze wurden die dazugehörigen Farben extrahiert, nach erfolgter Dilatation wurde ein spezifisches Region-Growing durchgeführt.

Grundlagen

Farbkanal Extraktion

Eine Farbkanal Extraktion ist eine simple Operation, welche die einzelnen Farbkanäle (Rot, Grün, Blau), eines Bildes extrahiert und separat zur Verfügung stellt. Mit diesen Daten können beispielsweise die roten Pixel in einem Bild gefunden werden.

Morphologische Operatoren

Morphologische Operatoren sind ein Teil von lokalen Operatoren. Lokale Operatoren beziehen für die Berechnung eines neuen Wertes eines Pixels nicht nur das zugrundeliegende Pixel bei, sondern auch seine Region.

In der Bildverarbeitung können morphologische Operationen auf Binär- und Graustufenbilder angewendet werden und sind für vielfältige Bildverarbeitungsaufgaben wie Kantenextraktion, Skelettierung oder Segmentierung geeignet. Beispiele eines morphologischen Operators sind die Dilatation oder Erosion, wobei Lücken von Pixelwerten geschlossen resp. geöffnet werden.

Hough Transformation

Die Hough-Transformation (HT) ist eine globale Transformation, die gebraucht wird, um geometrische Strukturen wie Geraden, Kreise und Ellipsen in einem bereits binarisierten Bild zu detektieren. Der Algorithmus geht auf Paul Hough zurück, der das Verfahren 1962 patentieren liess. Wie bei allen globalen Transformationen üblich, ist auch die Hough-Transformation in ihrer einfachen Variante eine rechenintensive Aufgabe. Positiv ist, dass sie sehr robust und wenig anfällig gegen Rauschen ist.

Region-Growing

Beim Region-Growing wird ein bestimmtes Pixel gewählt, von diesem aus werden seine Nachbarn analysiert. Wenn die Nachbarn ein bestimmtes Kriterium erfüllen (z.B. Schwellwert) werden sie zur Region hinzugefügt. Für die hinzugefügten Pixel werden wieder die Nachbarn gewählt und analysiert usw.

Vorgehen, Methoden, Analysen

Wir haben unser Vorgehen auf den bereits erlernten Grundlagen in der CPVR-Vertiefung des Studiums aufgebaut. Dabei wollten wir die Aufgabe nicht mit Hilfe einer Ähnlichkeitssuche von Walters Bild lösen. Wir haben uns entschieden wichtige Merkmale von Walter durch die Anwendung von lokalen und globalen Operatoren zu finden. Dabei ist wichtig zu erwähnen, dass es dadurch nicht möglich ist Walter anhand eines einzigen Merkmals zu finden. Deshalb wurde der Ansatz gewählt die Brille durch eine Kreis-Hough Transformation zu finden und Pullover/Mütze durch das horizontale rot-weiss Muster.

Brille

Um eine einfachere Entwicklung des Algorithmus zu ermöglichen, wurde mit einem Ausschnitt des Bildes gearbeitet, welcher wie folgt gewählt wurde:



Abbildung 1: Wembley Stadion

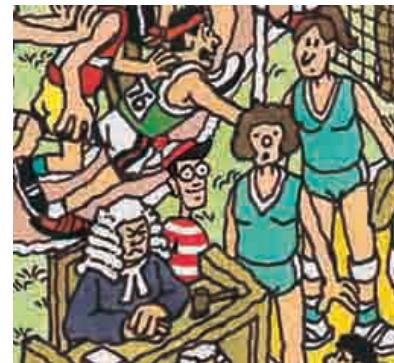


Abbildung 2: Ausschnitt Brillendetektion

Brillenglasränder

Die erste Idee war die Brille durch eine Extraktion der Brillenglasränder zu finden.

Dafür wurden zuerst die (schwarzen) Pixel, bei welchen alle 3 RGB Kanäle Werte kleiner als 0.3 (Hex 0x4C) besitzen, extrahiert.



Abbildung 3: Extraktion der schwarzen Farbe

Die Ränder der Brillengläser sind leider nicht nur schwarz, sondern gehen auch in einen Braunton über. Um die Lücken im Brillenrand zu schliessen wurde ein Closing Operator angewandt; dies bedeutet eine Dilatation gefolgt von einer Erosion.



Abbildung 4: Closing Operation

Danach wurden mit dem Canny Algorithmus, als Vorbereitung für die Hough Transformation, die Kanten extrahiert. Dies ist notwendig um bei einem ausgefüllten Objekt nicht n-Kreise, sondern nur derjenige, der aussen an der Kante liegt zu finden.



Abbildung 5: Canny edge detection

Man sieht bereits im Hough Raum, dass mit sehr vielen nicht eindeutigen erkannten Kreisen gerechnet werden muss.

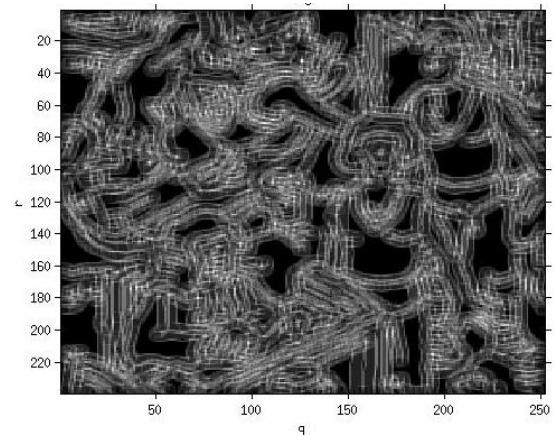


Abbildung 6: Hough Raum

Mit der Hough Transformation wurde nach Kreisen mit einem Radius von 3px gesucht. Dies entspricht dem Radius des rechten Brillenglasrandes. Dieser Kreis ist jedoch schon in diesem Bild erst unter den besten 150 Kreisen auszumachen.

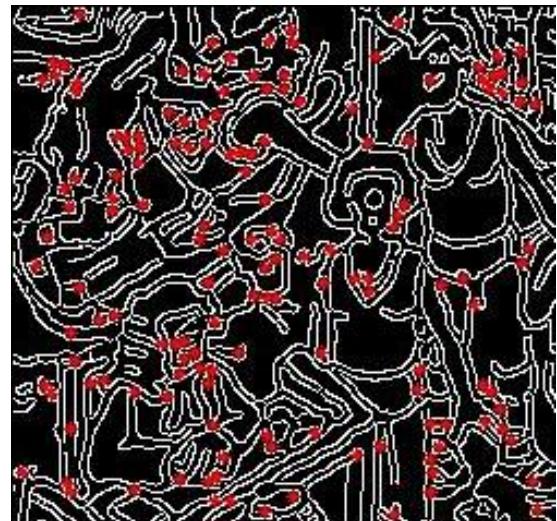


Abbildung 7: Gefundene Kreise mit Radius 3px

Es steht fest, dass mit Hilfe der Extraktion der schwarzen Farbe und dem Suchen nach den Rändern der Brillengläser, die Brille nicht gefunden werden kann. Dies hat zwei massgebliche Gründe:

- Da die Farbe Schwarz sehr oft gebraucht wird, unter anderem um Ränder/Kanten zu gestalten, sind zu viele Linien und somit mögliche Kreise übrig um nach der Brille zu suchen.
- Die Brillenglasränder sind nicht perfekt und enthalten auch noch Brauntöne. Diese Lücken können zwar mit einem Closing-Operator ziemlich gut geschlossen werden, jedoch nicht komplett. Dies führt unweigerlich zu schlechteren Ergebnissen bei der Hough Transformation.

Augapfel

Die die Idee, die Brillen von Walter mit Hilfe des Brillenrandes zu finden, gescheitert ist, wurde versucht nicht direkt die Brille, sondern seine markanten weissen Augen zu finden.

Dafür wurden zuerst die (weissen) Pixel, bei welchen alle 3 RGB Kanäle Werte grösser als 0.85 (Hex 0xD9) besitzen, extrahiert.



Abbildung 8: Extraktion der weissen Farbe

Die daruch entstanden "Lücken" im Augapfel, welche durch die schwarze Pupille oder dunklere Farbe verursacht werden, müssen geschlossen werden. Dafür ist eine Dilatation die geeignetste Operation.



Abbildung 9: Dilatation

Danach wurden auch wieder mit dem Canny Algorithmus, als Vorbereitung für die Hough Transformation, die Kanten extrahiert.

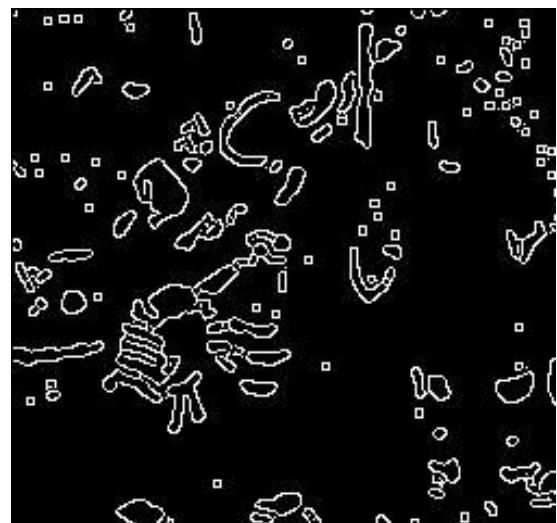


Abbildung 10: Canny edge detection

Der Hough Raum zeigt bereits einige eindeutige Kreise, unter anderem auch jene beim Augapfel (helle weisse Punkte).

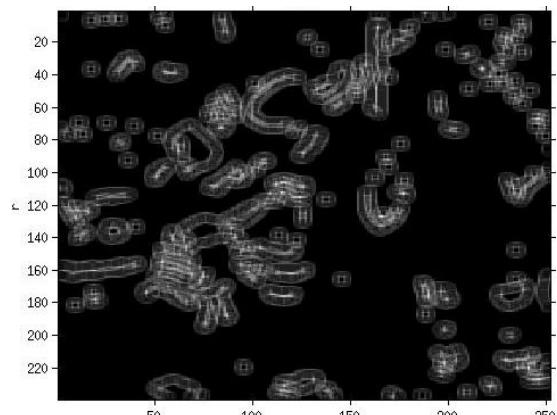


Abbildung 11: Hough Raum

Es wurden Kreise mit einem Radius von 3px (rot) sowie 4px (blau) gesucht um beide Augäpfel zu finden. Die beiden Augäpfel sind bereits Teil der besten 5 Kreisen.

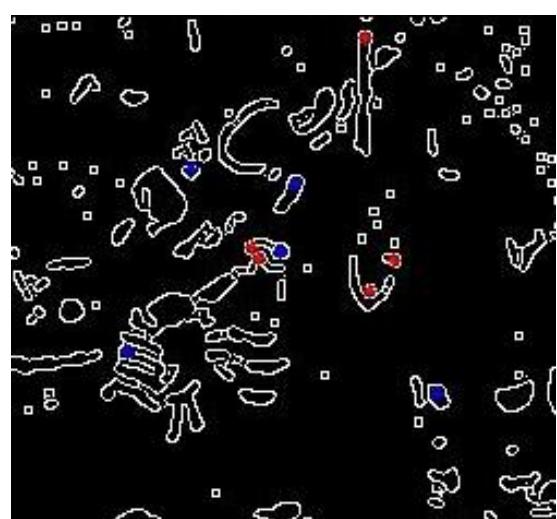


Abbildung 12: Gefundene Kreise mit Radius 3px (rot) und Radius 4px (blau)

Die gefunden Resultate können noch einge-grenzt werden, da die beiden Augäpfel eine gewisse Distanz zu einander haben.

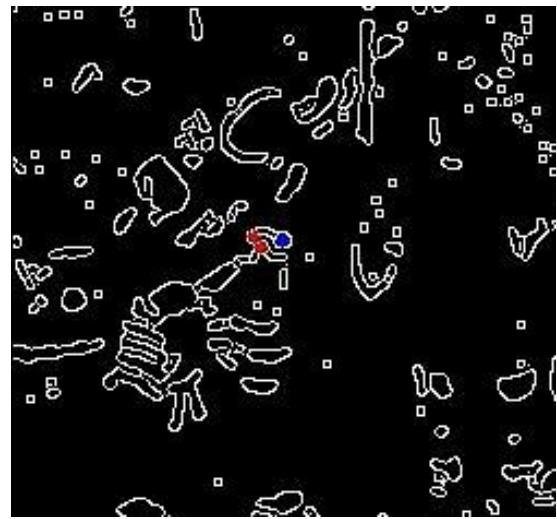


Abbildung 13: Reduziert auf rot-blaue Kreise mit bestimmter Distanz:
 $8px < Distanz < 16px$

Mit dieser Methode sind in diesem Bildausschnitt die Augäpfel von Walter eindeutig zu identifizieren. Um zu verifizieren, dass dies nicht nur in diesem Bildabschnitt funktioniert, wurde die Methode auf das ganze Bild angewandt. Dabei wurden unter Berücksichtigung der besten 200 Kreisen 8 mutmassliche Augapfel Paare gefunden. Diese bilden den Rückgabewert der Brillen-Analyse.

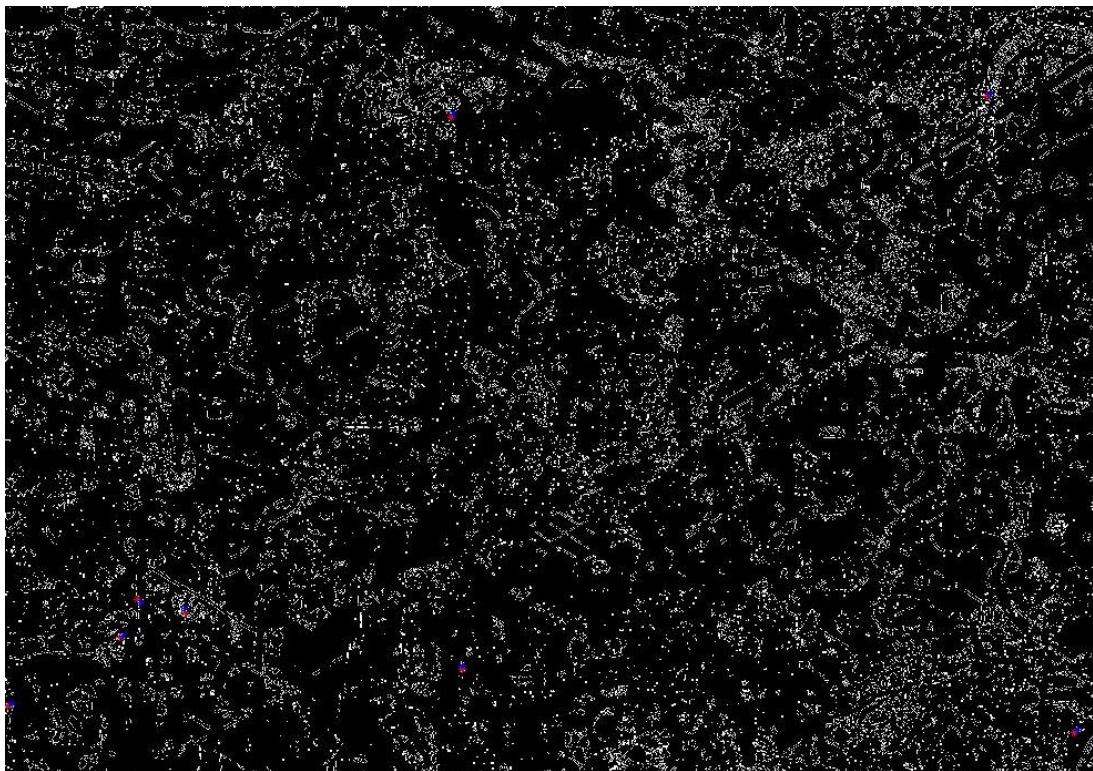


Abbildung 14: 8 Mutmassliche Augapfel Paare

Pullover/Mütze

Da Walters Pullover/Mütze aus weissen und roten Streifen besteht, wurde versucht einen zusammenhängenden Bereich aus roten und weissen Teilen zu finden.

Zuerst wurden die weissen Pixel extrahiert. Das heisst die Pixel, bei welchen alle 3 RGB Kanäle Werte grösser als 0.8 besitzen.



Abbildung 15: Extraktion der weissen Farbe

Anschliessend wurden die roten Pixel extrahiert. Das heisst die Pixel, bei welchen der Rotkanal doppelt so gross ist, wie das Maximum des Grün- und Blaukanals.



Abbildung 16: Extraktion der roten Farbe

Im nächsten Schritt wurden die roten und weissen Pixel zusammengefügt.



Abbildung 17: Extraktion der roten und weissen Farbe

Anschliessend wurde das Bild dilatiert um die Lücken zwischen den Streifen zu schliessen.



Abbildung 18: Dilatation

Im letzten Schritt wurde mittels Region-Growing Flächen gesucht, die rote und weisse Pixel enthalten. Zu diesem Zweck wurde über das Bild geloopt. Sobald ein rotes oder weisses Pixel gefunden wurde, wurde von diesem aus das Region-Growing gestartet. Dabei wurden die roten und weißen Pixel zur Fläche hinzugefügt. Für die hinzugefügten Pixel wurden wieder deren Nachbarn überprüft. Sobald eine Fläche vollständig gefunden wurde, wurde überprüft ob die Fläche sowohl rote als auch weiße Pixel enthält. War dies der Fall wurde die Fläche zum Resultat-Bild hinzugefügt, ansonsten wurde die Fläche verworfen.



Abbildung 19: Verbundene weiss/rote Fläche

Auf das ganze Bild angewandt ergibt sich folgendes Bild.

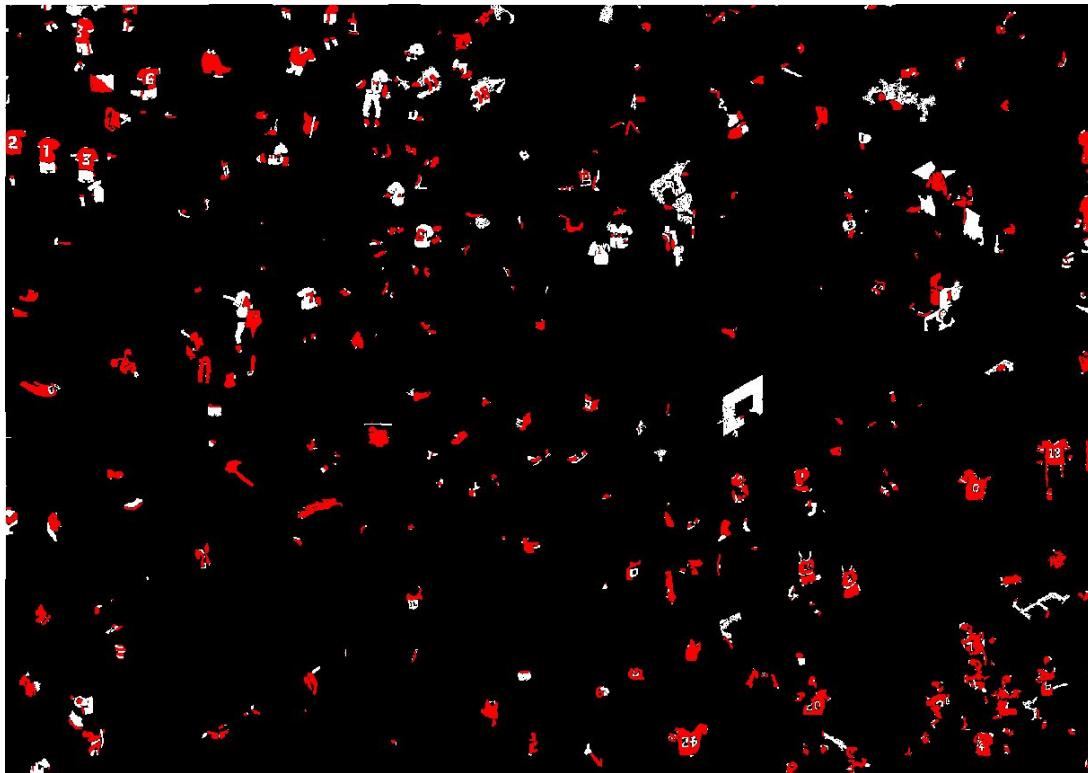


Abbildung 20: Verbundene weiss/rote Fläche

Mit der Matlab-Methode *regionprops* wurden von den rot/weissen-Flächen die Zentren extrahiert. Diese bilden den Rückgabewert der Pullover/Mütze-Analyse.

Brille und Pullover/Mütze zusammenfügen

Nach der Detektion von Brille und Pullover/Mütze wurden die beiden Daten zusammen gefügt.

Für jedes mutmassliche Augenpaar (Brille) wurden die mittels Matlab-Methode *knnsearch* die nächsten 2 rot/weissen Flächen (Mütze und Pullover) gesucht. Dort wo der Abstand zwischen Augenpaar und den beiden Flächen am kleinsten war, wurde Walter vermutet. Mittels Matlab-Methode *plot* wurde vom Augenpaar aus ein Kreis gezeichnet um den Fund zu markieren.



Abbildung 21: Mutmasslicher Walter

Wird der Algorithmus auf das ganze Bild angewandt, wird Walter ebenfalls gefunden.

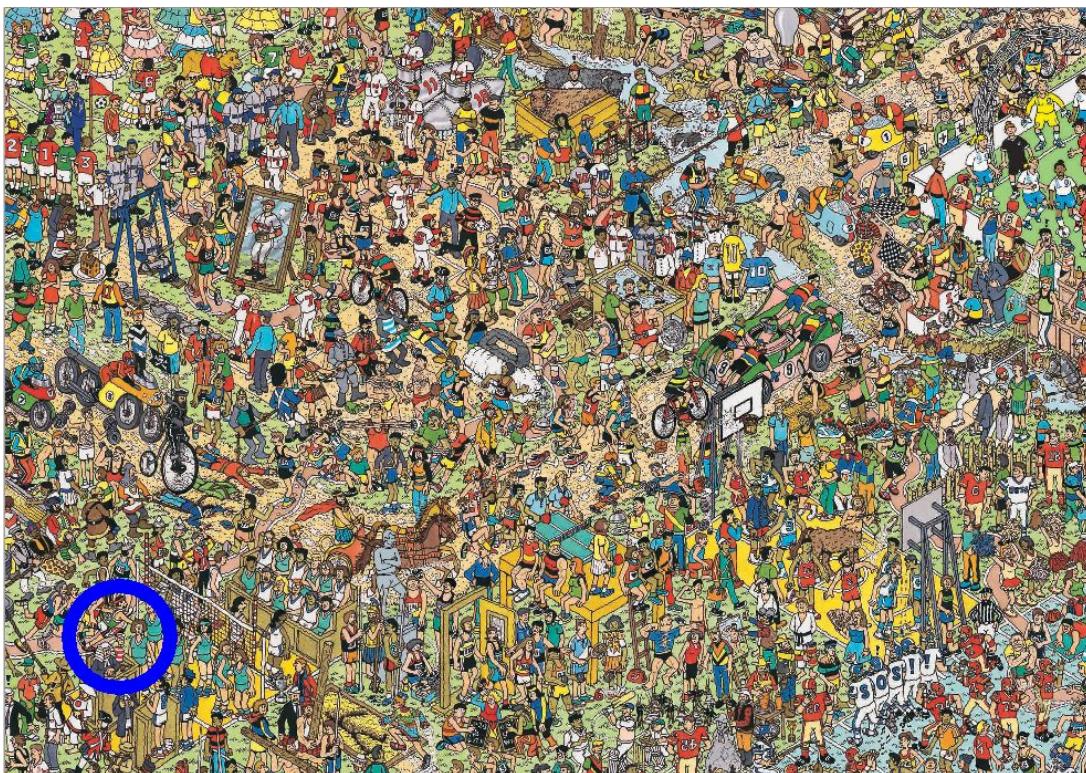


Abbildung 22: Mutmasslicher Walter auf dem gesamten Bild

Ergebnisse, Resultate

Wie in Abbildung 22 sichtbar wurde das Ziel Walter zu finden erreicht. Die im Konzept beschriebenen Methoden haben ausgereicht um im verwendeten Bild Walter zu finden. Für einen bildübergreifenden Algorithmus müsste die Lösung noch optimiert werden. Die Verwendung einer Ähnlichkeitssuche würde wahrscheinlich in verschiedenen Bilder bessere Resultate bringen, jedoch wurde diese in der Arbeit explizit ausgeschlossen.

Projekt-Repository

Das vollständige Programm zu diesem Projekt finden Sie auf GitHub:

<https://github.com/flec/Wally>

Starten Sie `/matlab/findWally.m`, um Walter zu finden.