

AUTOMATIC RUBIKS CUBE SOLVER FOR ANDROID

DAVID KÖRNER



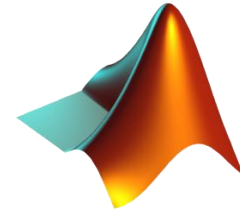
THEMA & ZIELE

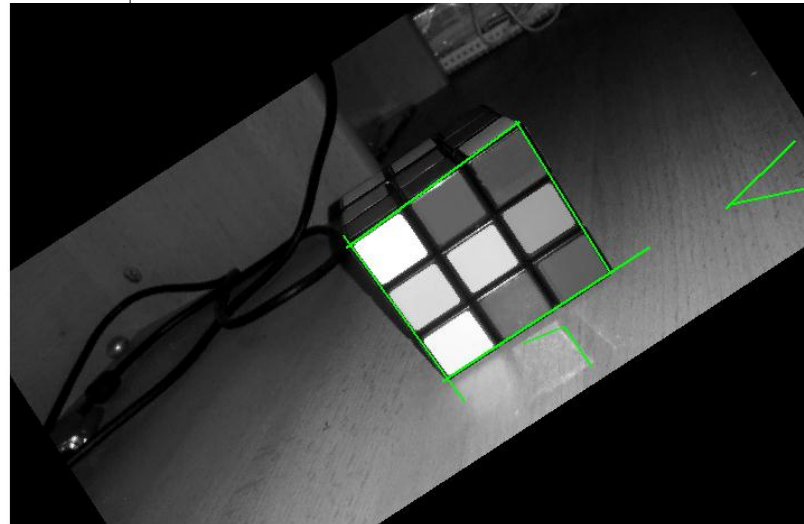
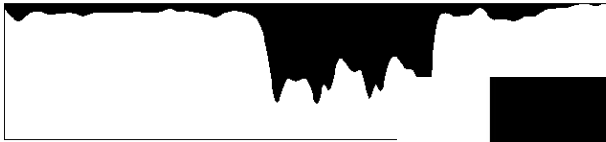
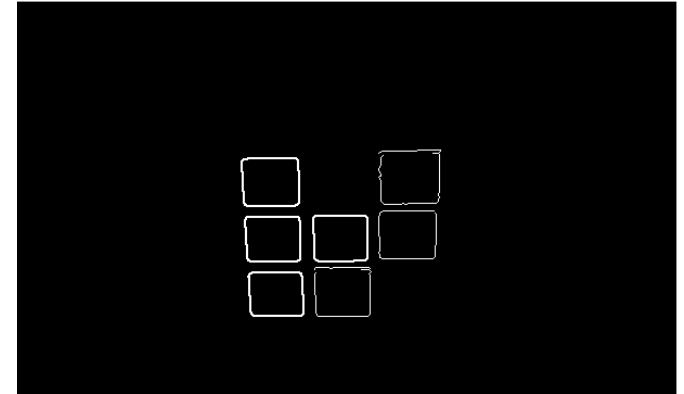
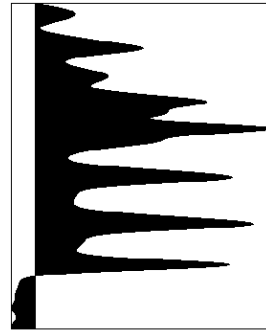
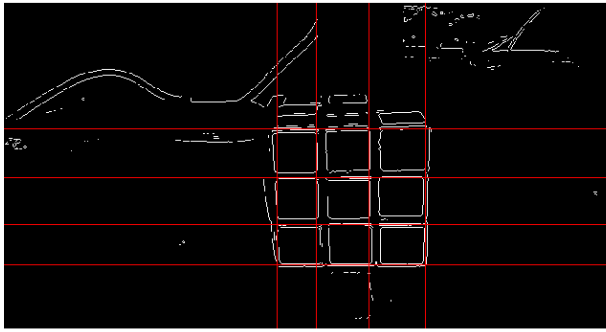
- **Szenario**
 - Hilfe beim Lösen des Würfels
 - Videoaufnahme des Würfels
 - Unterstützung beim Lösen durch CV
- **Teilbereiche**
 - Detektion Cube/Face/Facet
 - Auslesen (Clustering) der Farben
 - Berechnung und Ausgabe der Lösung



UMSETZUNG

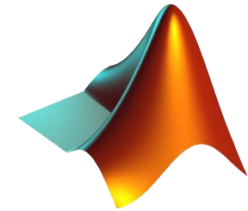
- **Prototyp**
 - Einfache Umsetzung in MATLAB
 - Verschiedene Ansätze testen
 - Finalen Lösungsansatz auswählen
- **Android Portierung**
 - Lösungsansatz als Android App umsetzen
 - Umsetzung mit Java/C++ Eclipse/Visual Studio
 - Verwendung von OpenCV, ADT, JNI





VERWORFENE ANSÄTZE

VORGEHENSWEISE



- **Prototyp in MATLAB**
 - Detektion des Würfels mittels
 - Standardabweichungsfiler + morph. Rekonstruktion
 - Einlesen der Faces
 - Ein Bild pro Face verarbeitet
 - Clustering der Farben
 - HSV Farbraum
 - „Sichere“ Zuweisungen zuerst

VORGEHENSWEISE



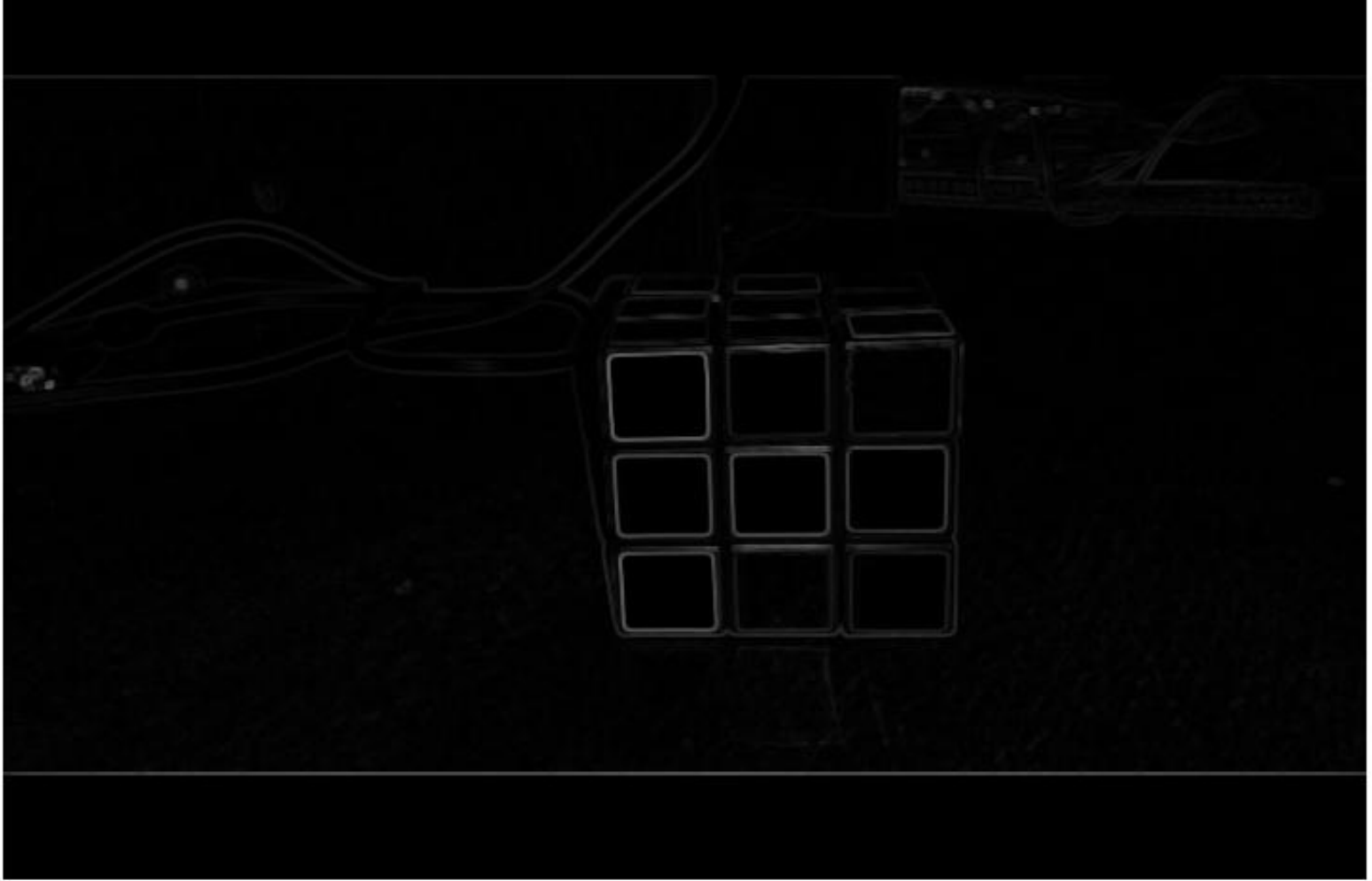
- **Android Portierung**
 - Detektion des Würfels mittels
 - Standardabweichungsfilter + Konturanalyse
 - Einlesen der Faces
 - Berührung des Screen startet Detektion eines Faces
 - Ende wenn 9 ~Quadrate gefunden
 - Weitergeben der Facets in geordneter Reihenfolge
 - Clustering der Farben
 - Farbe von Facet Mittelpunkten auslesen
 - Konvertieren in HSV Farbraum
 - tbd

PROBLEME

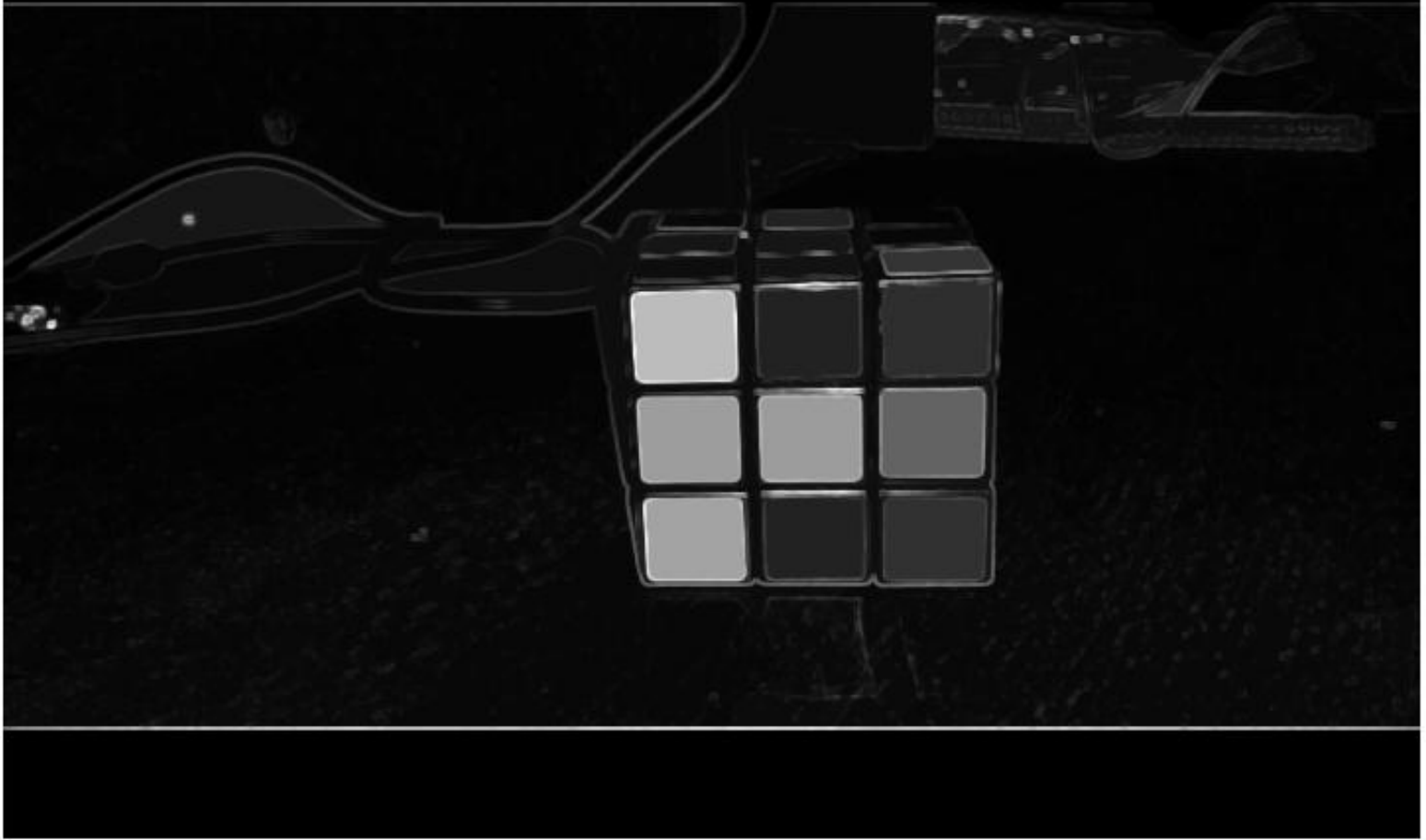
- **Aufwändige Einrichtung der Entwicklungsumgebung**
 - Eclipse + Android + OpenCV + JNI
- **Fehlende Erfahrung mit Technologien**
- **Mangel an Debugging Möglichkeiten**
- **keine OpenCV Umsetzung von ‚imFill‘**
 - Vereinfachter Lösungsansatz
 - Überarbeiten der Methode
- **Veränderung von Kamera ISO/Focus**
- **Kontrast bei dunkler Umgebung/Facet**



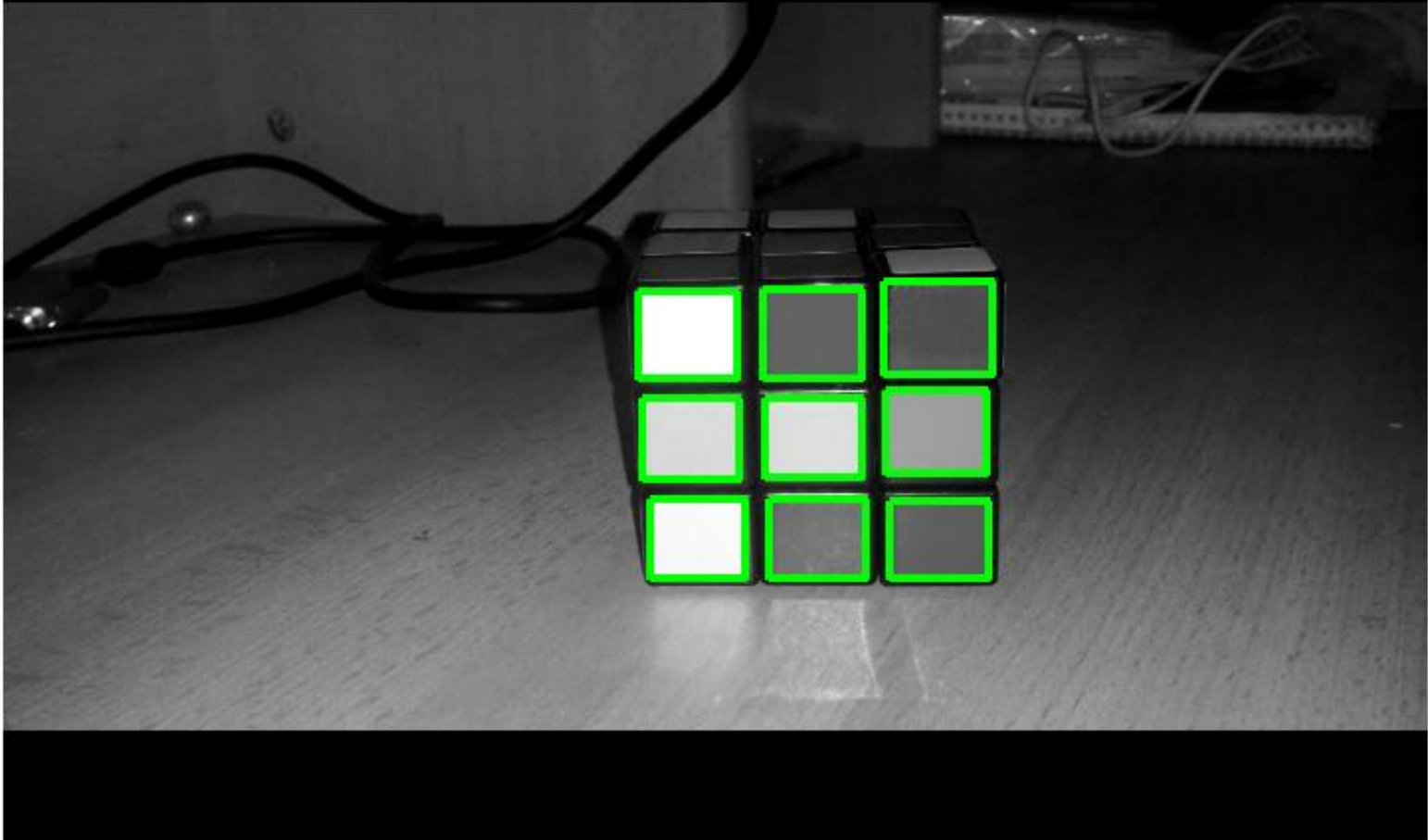
ORIGINALBILD



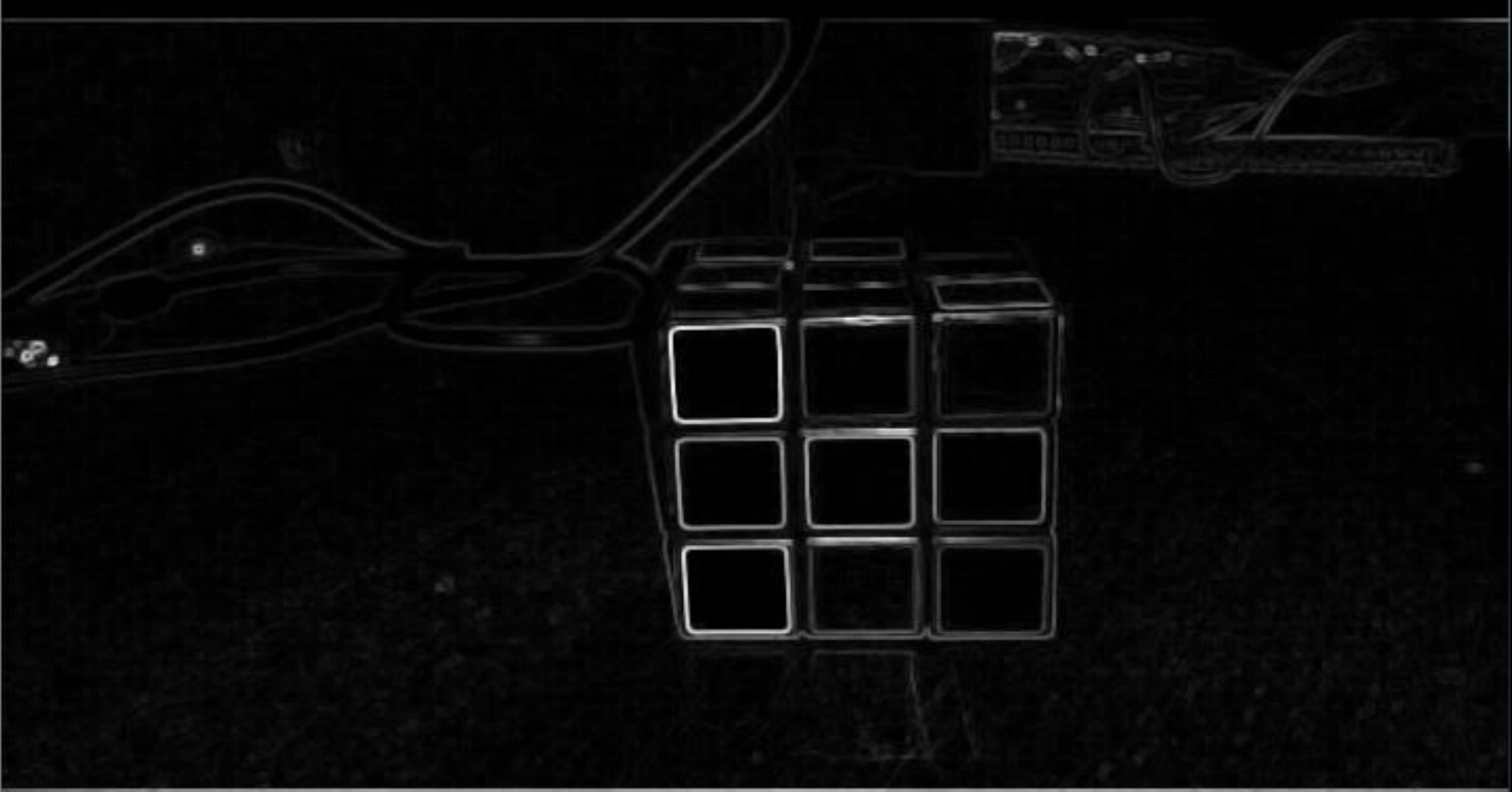
STDFILT (+ IMFILL) 1/3



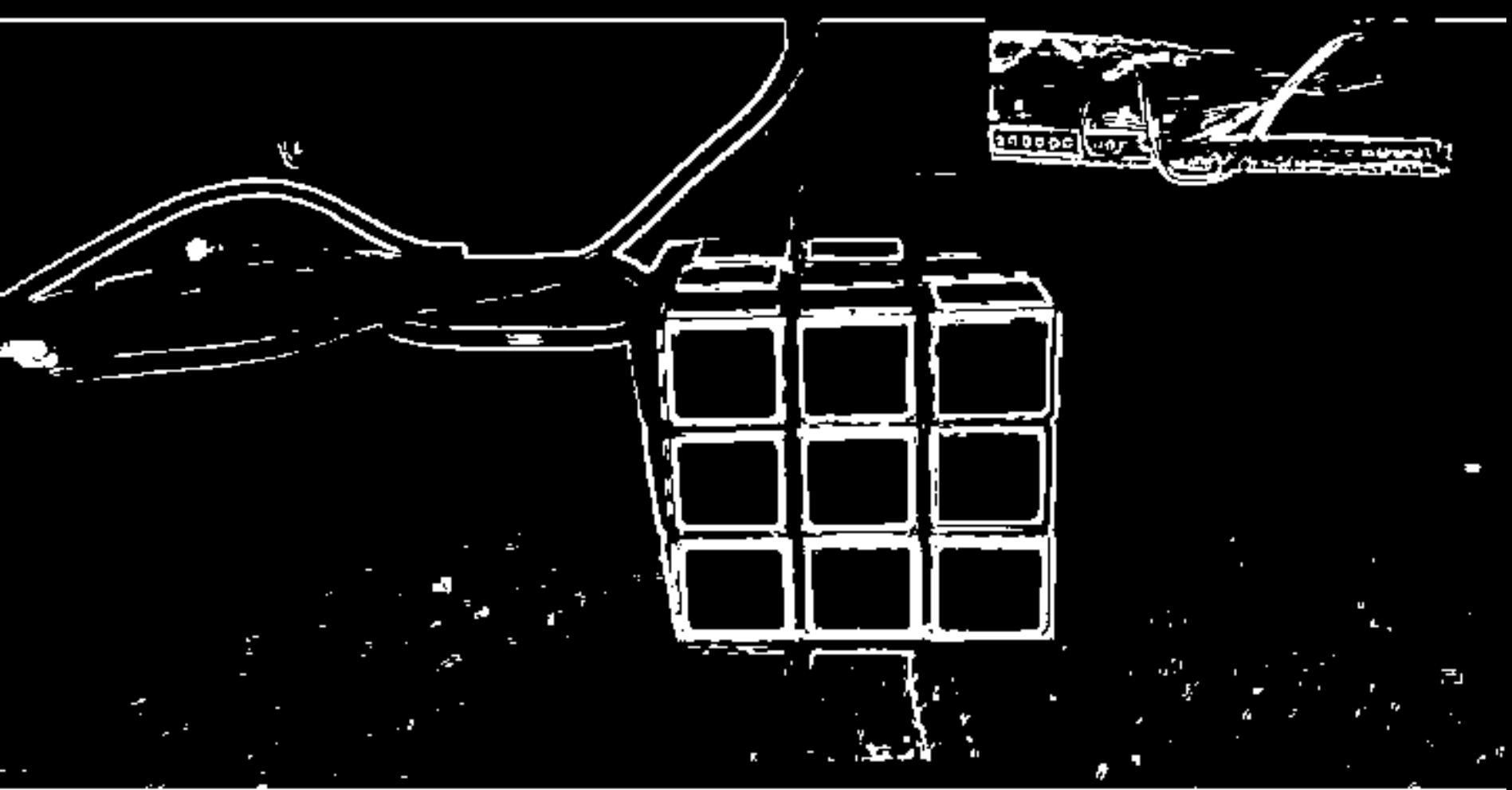
STDFILT + IMFILL 2/3



STDFILT + IMFILL 3/3



OPENCV: STDFILT (+ CONTOURS) 1/2



OPENCV: STDFILT (+ CONTOURS) 2/2

CONCLUSIO

- **Einarbeitungsaufwand nicht unterschätzen**
- **Einfache Lösungen anstreben**
- **Automatic Rubik's Cube Solver umsetzbar**
- **CV Algorithmen auf Android umsetzbar**
 - Vielfältige Einsatzmöglichkeiten
 - Prototyping mit MATLAB sinnvoll

**DANKE FÜR DIE
AUFMERKSAMKEIT!**