

TSBB11 - Bilder och grafik

OBS - Observatör av BridgeSpel



Peter Johansson
Johan Lorentzon
Fredrik Svensson
Ebba Wimby Schmidt

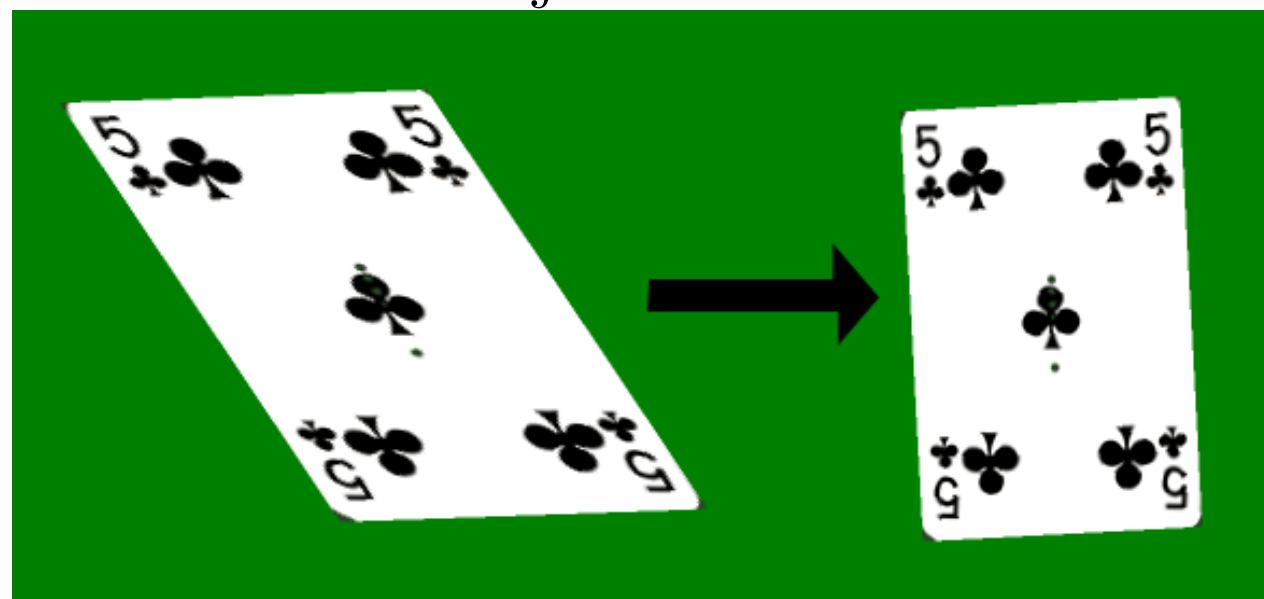
Bridge är ett populärt kortspel för fyra spelare. Hittills har protokoll från matcherna förts manuellt. OBS - Observatör av BridgeSpel - är en mjukvara för att automatisera detta med hjälp av bildbehandling. Målet var att korrekt kunna klassificera:

1. spelkort,
2. budkort och
3. bridgebrickor.

OBS-systemet använder en webbkamera som monteras ovanför bridgebordet. Mjukvaran är skriven i C++ och använder sig av mjukvarubiblioteket OpenCV.

Perspektivjustering

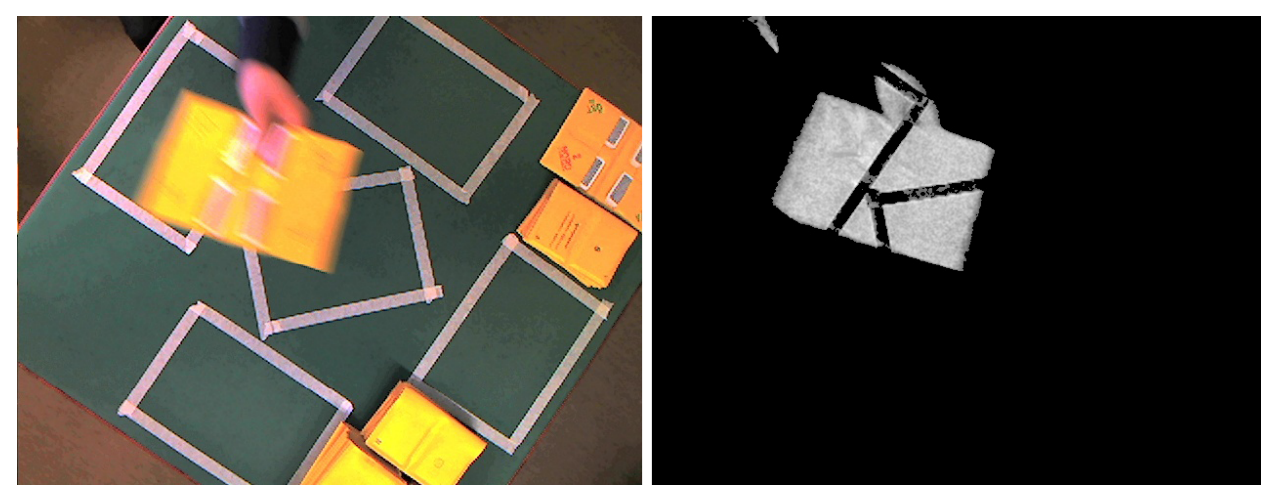
På bordsytan definieras fem utläggningsytor där föremål som ska tolkas kan läggas. Allt utanför dessa områden ignoreras. Genom att ansätta en hålkameramodell och att anta att bordets ytor är helt plana, kan bilden av objekten rekonstrueras.



Korrektion för perspektiv.

Rörelsedetektion

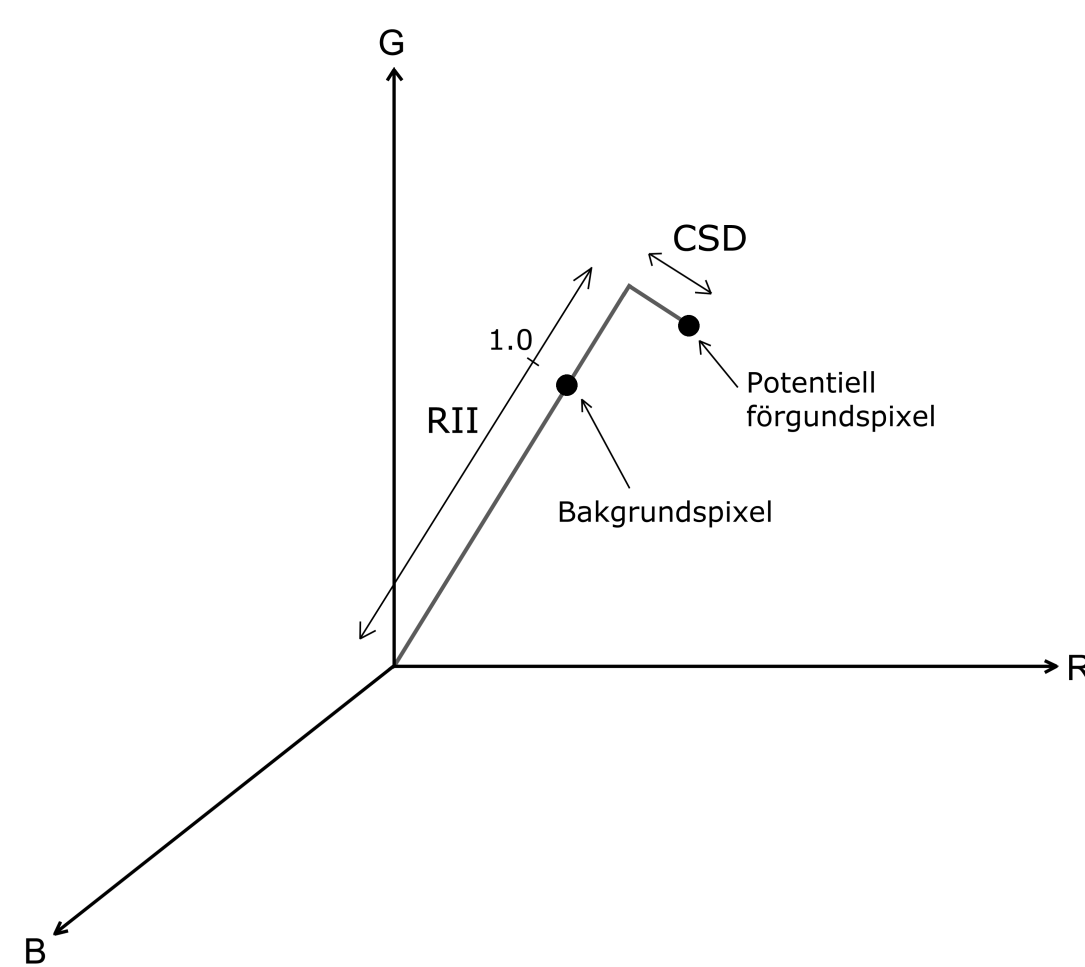
OBS försöker inte att klassificera objekt under tiden som bilden innehåller mycket rörelse. Mängden rörelse skattas genom att jämföra scenen med en bakgrundsmodell som kontinuerligt uppdateras med ett approximativt medianfilter.



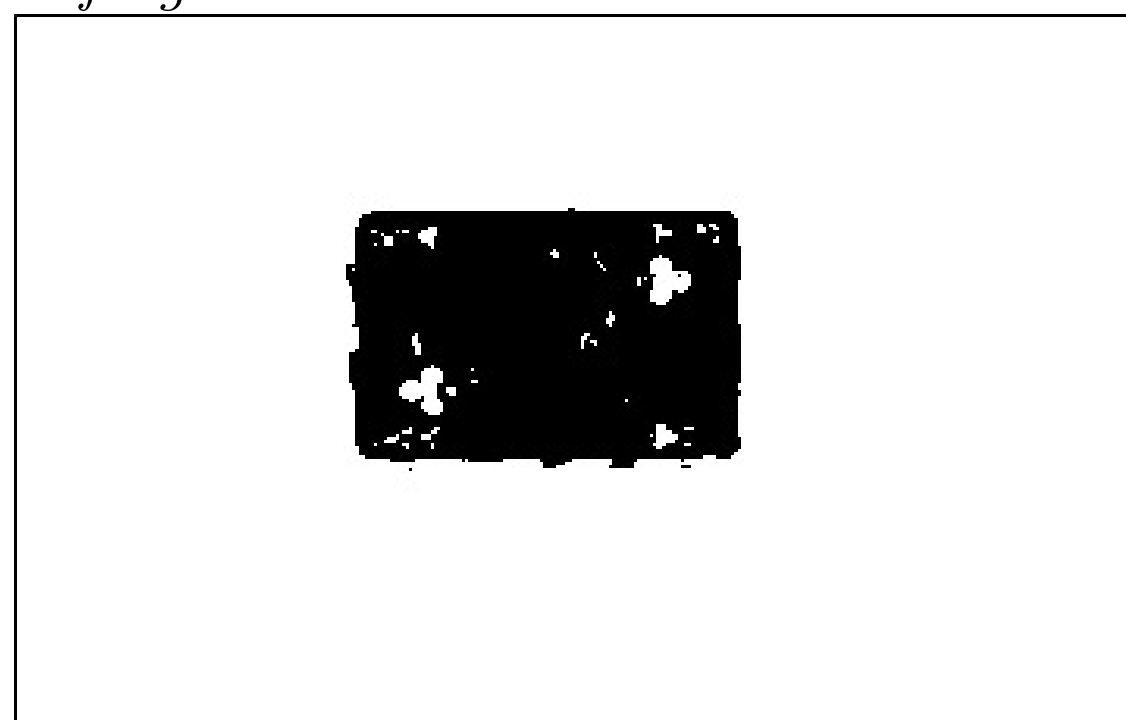
Mängden rörelse när en ny bricka läggs på bordet.

Segmentering av förgrund/bakgrund

Ytterligare en bakgrundsmodell används för att detektera förgrund i bilden. För att undvika falska positiva svar vid skuggor används en färgrymd som klarar att kompensera för skuggor under flera olika ljusförhållanden.



Färgrymden som används för att segmentera fram förgrunden.



Detekterad förgrund.

Objektlokalisering

Som ett första steg används morfologisk öppning följt av slutning för att ta bort enskilda falska positiva pixlar samt håligheter i objekt. Vi anpassar en rek-

tangel till det största sammanhängande föremålet. Vi föreslår följande algoritm:

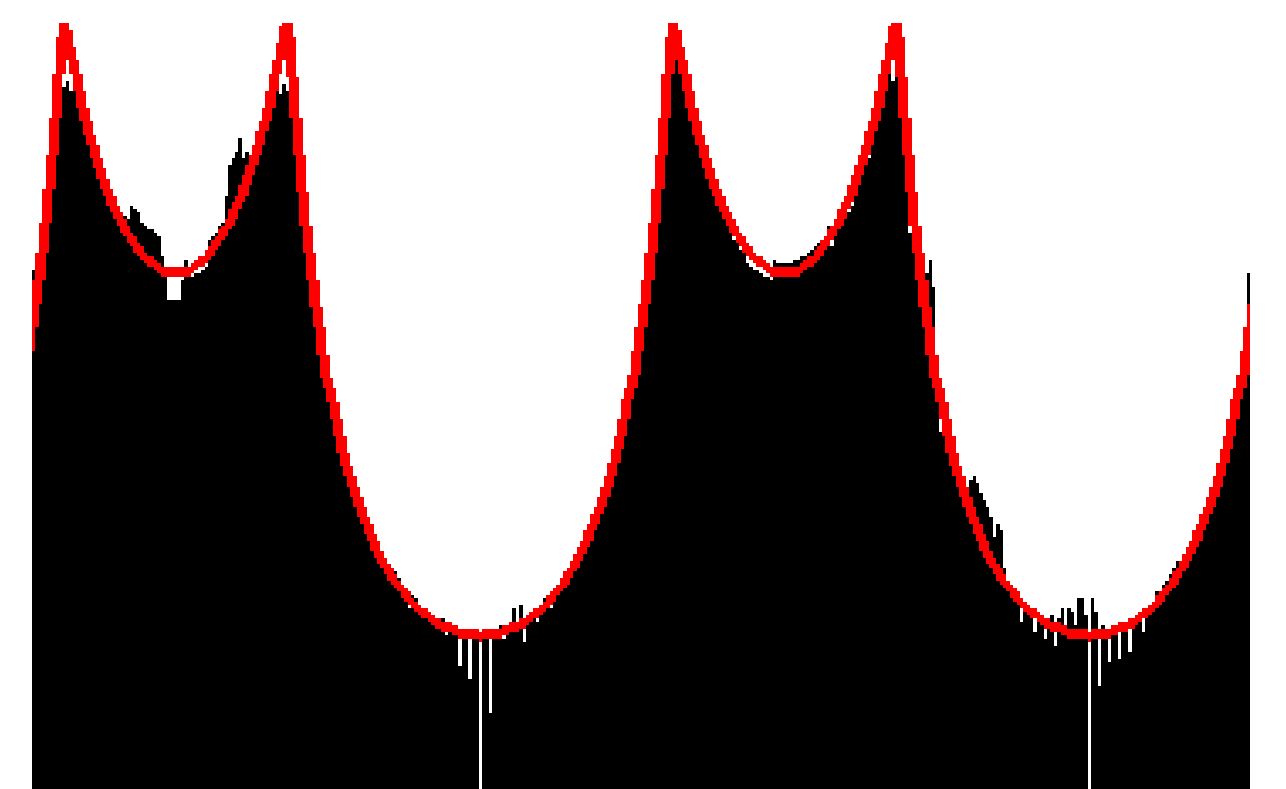
1. Beräkna avståndet från tyngdpunkten till den pixel som är längst bort, i varje vinkel. Lågpassfiltrera. Detta är konturen i polära koordinater, $R(\phi)$.
2. $\phi_1 = \arg \min R(\phi) + R(\phi + 180^\circ)$ utgör vinkeln till kortsidorna.
3. Bredd och höjd kan nu skattas som

$$b = \frac{2}{2\Delta + 1} \sum_{\theta=-\Delta}^{\Delta} \cos(\theta) R(\phi_1 + \theta)$$
$$h = \frac{2}{2\Delta + 1} \sum_{\theta=-\Delta}^{\Delta} \cos(\theta) R(\phi_1 + 90^\circ + \theta)$$

4. Utifrån skattad bredd och höjd kan en $R_{ideal}(\phi)$ beräknas. Skattningen av rotation förfinas genom att hitta

$$\phi_2 = \arg \min_{\phi} \sum_{\alpha} (R_{ideal}(\alpha) - R(\alpha - \phi))^2$$

Endast ett fåtal värden runt ϕ_1 behöver provas.



En ideal rektangel anpassas till uppmätta värden i polära koordinater.

Klassificering

De urklippta bilderna jämförs med en databas av kända objekt med hjälp av template matching. För att snabba upp proceduren görs matchningen först i en låg upplösning. De bästa kandidaterna går sedan vidare till att matchas i hög upplösning.