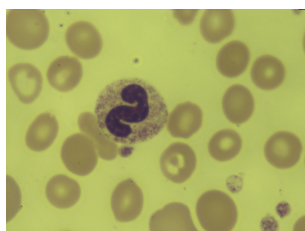


# Förlustfri komprimering och högkvalitativ förstörande komprimering av blodcellsbilder för bildvisning och maskininlärning

Terese Nilsson Daniel Hamngren

2013-06-13

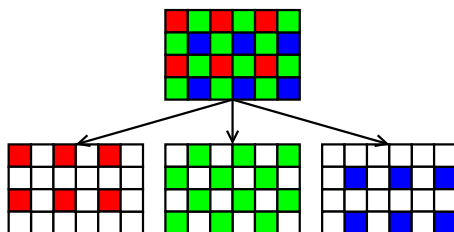
För att analysera blod kan CellaVisions analysverktyg användas. De utför analys av blod med hjälp av mikroskopibilder på vita blodceller. Cellerna klassificeras automatiskt och fördelningen av de olika typerna av celler kan användas för att diagnostisera patienten. Dessa bilder behöver, för att kunna sparas effektivt, komprimeras.



Figur 1: Exempel på cellbild från analysverktygen.

Vårt projekt handlar om att studera och utveckla olika metoder för att komprimera bilder på blodceller. Vi har främst tittat på förlustfri komprimering (ingen information från bilden förloras) men även förstörande komprimering (information från bilden förloras).

Eftersom komprimeringsmetoden endast ska användas för bilder på blodceller från CellaVisions analysverktyg, kan den göras väldigt specifik. Kamerorna som tar bilderna har ett så kallat Bayer-filter monterat framför sensorn. Det gör att varje pixel i bilden direkt från kameran, *mosaikbilden*, endast innehåller information om en av färgerna röd, grön och blå. För att få fram en vanlig RGB-bild med tre värden för varje pixel interpolerar man fram de saknade färgkanalerna för varje pixel. Genom att vi vet hur filtret ser ut och att interpolationen som används är gjord så att det går att åter skapa mosaikbilden exakt, kan vi spara bilderna en tredjedel så stora genom att endast spara mosaikbilden. Mosaikbilden kan delas upp i tre färgkanaler på följande sätt:



Figur 2: Uppdelning av mosaikbilden i färgkanaler.

De tomma pixlarna i färgkanalerna kan tas bort eftersom de inte innehåller någon information. Den gröna färgkanalen kan även delas upp i två färgkanaler.

## Förlustfri komprimering

Vi har testat flera olika metoder för att komprimera bilderna förlustfritt. Gemensamt för dem är att vi utgår från mosaikbilden och delar upp den i åtminstone tre separata färgkanaler. Varje färgkanal komprimeras sedan med någon förstörande metod. Skillnaden mellan varje pixel i den komprimerade kanalen och den ursprungliga färgkanalen räknas ut och sparas i en så kallad *residualbild*. Residualbilden komprimeras ytterligare med Huffmankodning som baseras på hur vanliga de olika värdena i bilden är. Vanliga värden sparas med kortare kodord och ovanligare värden med längre. Detta gör att residualbilden tar mindre plats att spara än om alla värden hade sparats med lika långa kodord. Den komprimerade bilden består av residualbilderna och de med förlust komprimerade kanalerna.

Metoden som gav bäst komprimeringsresultat är baserad på den statistiska modellen ARX. Mosaikbilden delas där upp i fyra lika stora kanaler, en röd, två gröna och en blå. Värdet på en pixel skattas med hjälp av pixlar före på samma rad och på raden ovanför. I alla färgkanaler utom en av de gröna används också information från de andra gröna färgkanalerna som hjälp i skattningen. Det skattade värdet på pixeln jämförs med det riktiga värdet och skillnaden sparas i en residualbild. Med den här metoden behöver vi endast spara fyra pixlar per färgkanal och de huffmankodade residualbilderna.

Okomprimerade är bilderna sparade med 24 bitar per pixel (bpp). När mosaikbilden är återskapad är de komprimerade ner till 8 bpp. Komprimeringsmetoden som använder ARX, beskriven ovan, ger ett resultat på 3,0409 bpp, vilket innebär att bilden nu är komprimerad till 12,67% av ursprungsstorleken.

## Förstörande komprimering

Den förstörande komprimeringsalgoritmen baseras på samma grundtankar som de förlustfria metoderna. Först delas mosaikbilden upp i fyra färgkanaler. Varje färgkanal komprimeras förstörande med den redan befintliga förstörande komprimeringsmetoden JPEG 2000 och residualbilder sparas. Skillnaden här är att förgrund (celler) och bakgrund hanteras olika. Eftersom bakgrunden inte är så viktig accepterar vi skillnader från originalbilden där. För- och bakgrunden separeras genom att hitta ett tröskelvärde. Genom att sätta värdet i residualbilden till noll för alla bakgrundspixlar, låter vi bakgrunden komprimeras förstörande, medan förgrunden fortfarande är förlustfritt komprimerad. Denna metod ger ett resultat på 2,4451 bpp.