

EXAMENSARBETE Filtered Path tracing using Halide**STUDENT** Fredric Berg**HANDLEDARE** Michael Doggett (LTH)**EXAMINATOR** Flavius Gruian (LTH)

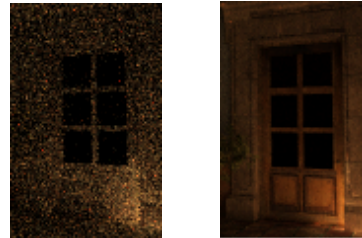
Framtiden för bildbehandling?

POPULÄRVETENSKAPLIG SAMMANFATTNING **Fredric Berg**

Halide är ett modernt programmeringsspråk designat för bildbehandling. Detta arbete utforskar dess styrkor och svagheter genom att använda det för optimeringen av en bildrekonstruktions algoritm.

Bildbehandling är en viktig del av dagens samhälle. Det används i många olika fält som datorgrafik, medicin och datorseende. Det största problemet som förhindrar fortsatt utveckling är hårdvara och exekveringstider. I mobila enheter är det speciellt viktigt, då hårdvaran är ännu mer begränsad. En applikation som är begränsad av exekveringstid är path tracing. Det är ett sätt att rendera fotorealistiska bilder av hög kvalitet. Det största problemet för sådana verktyg är att renderingstiden är väldigt lång. Genom att ha väl optimerad kod kan man sträcka gränserna för vad som är möjligt ännu längre. Med utvecklingen av hårdvara och större fokus på optimering skulle det kunna bli en möjlighet att utnyttja path tracing eller liknande metoder för att rendera bilder i realtid. Halide är ett programmeringsspråk som är designat för bildbehandling. Principen bakom språket är att det ska bli lättare att optimera kod jämfört med andra språk som används i dagsläget. I språk som C++ är det krångligt att skriva effektiv kod och den blir också svår att tolka för oinsatta. I Halide kan en optimerad version skrivas med enklare kod och med mycket bättre överblick. I det här examensarbetet implementeras en bildrekonstruktions algoritm med hjälp av Halide för att undersöka möjligheterna och begränsningar med språket. När man renderar en bild med path tracing så finns det mycket information att hämta från processen. Ju längre man renderar desto hö-

gre kvalitet får den slutgiltiga bilden. Genom att köra en snabb rendering får man en lågkvalitativ bild. Algoritmen som presenteras i detta arbete tar informationen från en snabb rendering och använder den för att återskapa bilden. På detta viset får man en hög kvalitet på den slutgiltiga bilden men framförallt mycket kortare renderingstid.



Arbetet visar att Halide inte är den ultimata lösningen för optimering av all typ av kod. På grund av begränsningar inom språket är det inte lämpligt att använda det för alla typer av algoritmer. Halide stödjer bland annat inte datastrukturer som träd, vilket är vanligt förekommande inom datorgrafik. Det medför problem som försämrar prestandan. För det här projektet blev det ingen förbättring av exekveringstid jämfört med en implementation i C++. Det kan istället fungera som en studie om vad Halide är lämpligt och olämpligt för. Det belyser det som gör Halide till ett effektivt språk och även problematiken i vissa användningsområden.