



NXP-2017-K1

Závěrečná zpráva

Průběh řešení projektu: 20.3.2017 - 31.7.2017

Ing. Roman Juránek, Ph.D.

Ing. Tomáš Milet

Implementace jednoduché pipeline pro Structure from motion

Cíle

Hlavním cílem bylo implementovat prototyp jednoduché pipeline pro odhad hloubky obrazu ze dvou po sobě následujících snímků a provozovat tento prototyp na cílovém zařízení (i.MX 6).

Metoda

Metoda se skládá ze dvou hlavních částí - 1/ odhad relativního pohybu kamery a 2/ odhad hloubky zájmových bodů ze dvou snímků (stereo).

První část lze řešit dvěma způsoby detekcí význačných bodů s deskriptory (SIFT, SURF, apod.) a jejich párováním (matching), nebo sledováním bodů v obraze (např. pomocí KLT trackeru). Význačné body a deskriptory jsou robustnější při rychlém pohybu v obraze, kdy se sledování pomocí KLT může ztratit. Relativní pohyb kamery (rotace a translace) pak lze získat z korespondujících dvojic bodů získaných ze dvou po sobě následujících snímků pomocí výpočtu esenciální matice a jejího rozkladu. Translační vektor značí pouze směr, jeho velikost (tedy rychlost pohybu) nelze získat pouze z vizuálních informací. Bez znalosti rychlosti, lze vzdálenosti bodu od kamery odhadnout pouze v relativním měřítku, pro metrickou rekonstrukci je potřeba dodat rychlost externě.

Druhá část - odhad hloubky zájmových bodů (rekonstrukce scény) - pracuje s odhadnutým relativním pohybem z předchozího kroku a projekcemi zájmových bodů v obrazech. Pomocí triangulace lze získat pozice bodů ve 3D prostoru a tím i jejich (relativní) vzdálenost od kamery.

Rozšířením popsané metody může být například inkrementální rekonstrukce, kdy se pohyb kamery zjišťuje relativně k již známé 3D struktuře. Zde je nutné používat *bundle adjustment*, který zpětně upravuje pozice kamer v předchozích snímcích tak, aby projekce bodů do obrazu byly co nejpřesnější a tím se tvořila kompaktní 3D struktura.

Detailní popis metod a algoritmů je v dodané technické zprávě.

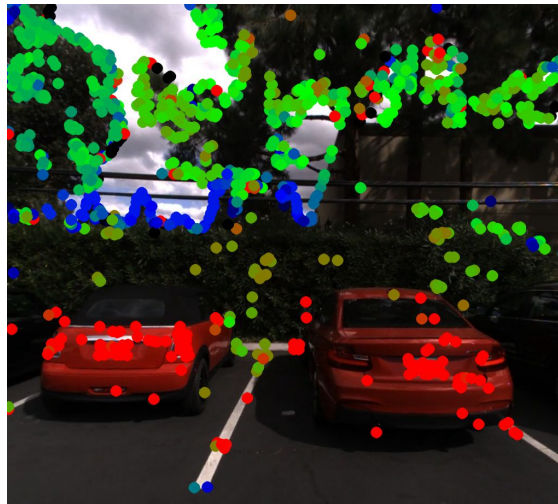
Průběh implementace

Byla implementována základní verze SFM v prostředí MATLAB, které nabízí širokou škálu prostředků pro práci s obrazem, detekci klíčových bodů a odhad struktury. Zde byly identifikovány potřebné algoritmy a postupy.

Na základě poznatků z MATLAB implementace byla vyvinuta C++ implementace pro cílové zařízení i.MX6. Tato implementace je postavena na sledování význačných bodů pomocí KLT trackeru a využívá OpenVX. V souvislosti s touto implementací byla nahlášena chyba implementace KLT trackeru v OpenVX.

C++ implementace je rozdělena na dvě části - akcelerovanou pomocí GPU a neakcelerovanou. Akcelerovaná část používá Harrisův detektor pro detekci rohů a sledování těchto bodů pomocí KLT trackeru. Akcelerace je realizována prostředky OpenVX. Neakcelerovaná část používá knihovnu OpenCV, nebo jsou algoritmy implementované přímo. Jedná se o část nalezení esenciální matice, stanovení pozice a triangulace bodů. K vizualizaci mračna bodů je využito OpenCV.

Další rozšíření implementace může spočívat v akceleraci CPU části algoritmu pro snížení výpočetního času. Nebo zvýšení přesnosti algoritmu pomocí metod jako je využití vícero snímků, bundle adjustment, apod.



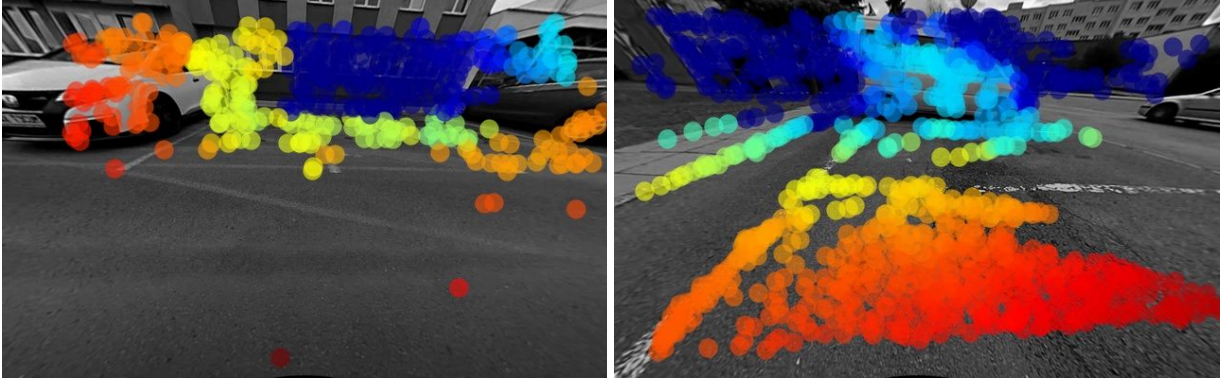
Vizualizace hloubky bodů na i.MX6

Poskytnutá data

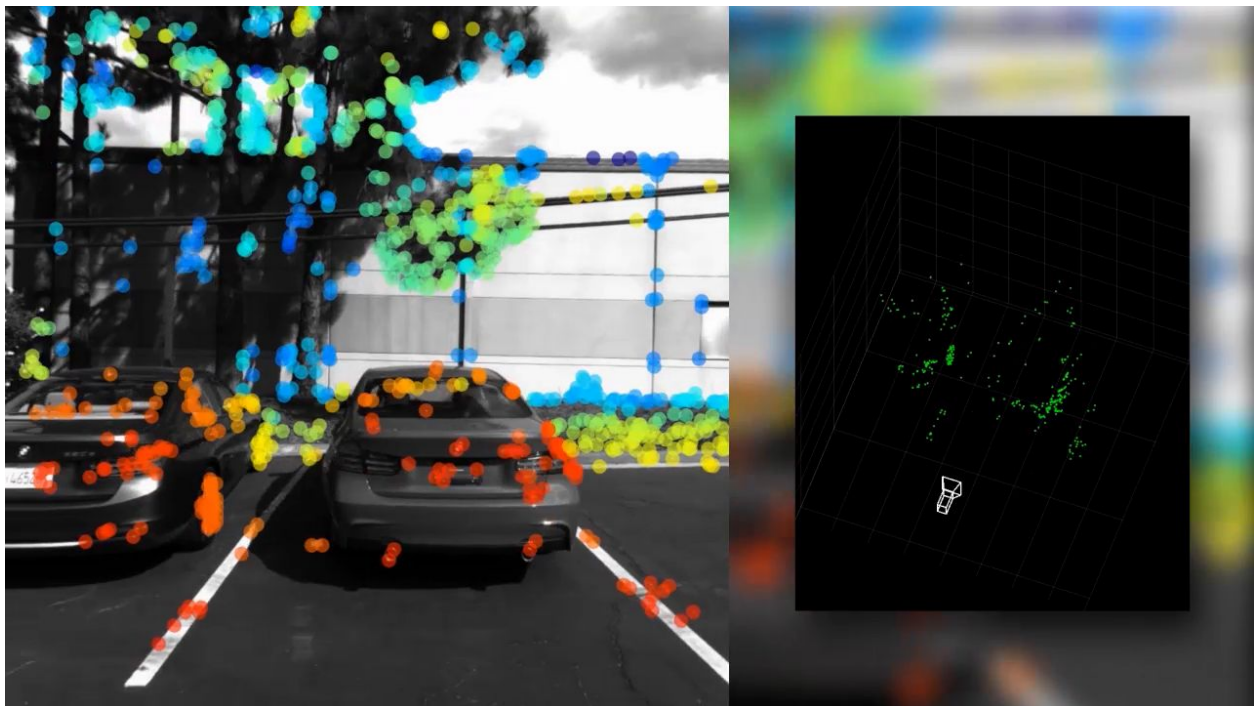
NXP poskytlo několik sekvencí pořizených z jedoucího auta kamerou směřující do boku. Na těchto sekvencích byla demonstrována funkčnost vyvinutých metod.

Dodané výsledky

- MATLAB implementace - zdrojové soubory pro SFM s použitím SURF a KLT trackeru.
- C++ implementace - zdrojové soubory pro SFM s použitím KLT trackeru pro i.MX6
- Technická zpráva s popisem metody
- Prezentační videa



Barevně kódovaná hloubka



barevně kódovaná hloubka zájmových bodů (vlevo), a 3D rekonstrukce (vpravo).

Závěr

V rámci projektu byl úspěšně implementován prototyp pipeline pro jednoduchý odhad hloubky ve videu pořízenem z jednocího auta. Byla dodána technická zpráva, základní implementace s použitím prostředků MATLAB a C++ implementace pro zařízení i.MX6 fungující v reálném čase (do 5 fps). Pro demonstraci bylo vytvořeno několik videí s vizualizací výsledků algoritmu.