

Kontrola kvality balení využívající 2D-3D matching

Software TH04010144-V2

Ondřej Klíma

Fakulta informačních technologií
Vysoké učení technické v Brně
Božetěchova 2, Brno 61266

Tento software TH04010144-V2 byl vytvořen s finanční podporou TA ČR.

Popis

Vytvořený software poskytuje sadu nástrojů pro kontrolu kvality balení produktu, konkrétně kontrolu vzájemného umístění produktu a balení. Vstupem pro kontrolu jsou snímky z obyčejné kalibrované RGB kamery, kontrola umístění ale probíhá v trojrozměrném prostoru. K dispozici je potřeba mít trojrozměrné modely produktu i balení, které je možné efektivně a levně získat pomocí optického 3D skenování. Modely jsou použity pro generování data setu virtuálních snímků jak pro trénování konvolučních neuronových sítí, určených pro segmentaci vstupních snímků [He2018] a zpřesnění detekce polohy, tak pro 2D-3D registraci [Klíma2016], která umožňuje odhad polohy produktu i balení v trojrozměrném prostoru pouze z kalibrovaného dvourozměrného snímku. Použití nástrojů bylo demonstrováno na případové studii balení kancelářských korektorů Tipp-Ex, jejíž výsledky jsou popsány v dokumentu Experimentální příprava algoritmů zpracování obrazu a implementace vybraného algoritmu v chytrém systému, příloze k průběžné zprávě projektu TH04010144 za rok 2020. Příklady správných a chybných balení jsou ilustrovány na Obrázku 1.



Obrázek 1: Ukázka balení korektorů z případové studie. Nástroje slouží k odhalení takových balení, která jsou ilustrována první dvěma kusy zprava.

Sada nástrojů obsahuje skripty a aplikace pro generování virtuálních data setů, nástroje pro jejich převod do požadovaných formátů využitelných pro segmentaci, nástroje pro trénování a použití segmentační sítě Mask-RCNN, pro 2D-3D registraci a pro trénování a použití regresní konvoluční neuronové sítě pro zpřesnění odhadu polohy.

Licence

Software je distribuován v open-source podobě pod licencí **OPEN SOURCE LICENCE VUT V BRNĚ**, jejíž úplné znění je k dispozici níže jako Příloha A.

Umístění

Skripty a zdrojové kódy jsou k dispozici volně ke stažení na adrese <http://www.fit.vutbr.cz/research/prod/?id=658>. Ukázková data případové studie kontroly umístění kancelářských korektorů Tipp-Ex v rámci zataveného balení jsou k dispozici ke stažení na <https://www.fit.vutbr.cz/~iklima/TH04010144-V2/>.

Prerekvizity

Implementační jazyky jsou Python 3 a C++ v kombinaci s kompilátorem Microsoft Visual C++ 2017 na platformě MS Windows.

Generování souboru dat pro trénování neuronových sítí

- [Blender](#)
- [pycocoreator](#)

Pro generování data setu syntetických snímků je potřeba mít k dispozici nainstalovaný volně dostupný modelovací nástroj [Blender](#), ideálně v aktuální verzi. Dále je potřeba mít k dispozici nainstalovaný Python balíček [pycocoreator](#), sloužící k vektorizaci binárních masek a jejich převod do [COCO](#) formátu.

Nástroje založené na konvolučních neuronových sítích

Nástroje pracující s konvolučními neuronovými sítěmi jsou implementované v jazyce Python a vyžadují instalaci balíčků:

- [Tensorflow](#)
- [Keras](#)
- [scikit-learn](#)
- [pandas](#)
- [PyTorch](#)
- [Detectron2](#)
- matplotlib, numpy, cv2, cython...

Trojrozměrná registrace modelů

- [Qt Toolkit](#)
- [OpenCV](#)
- MultiFragmentRegister library

Nástroje pro 2D-3D registraci využívají [Qt Toolkit](#), sloužící jako platformě nezávislé vývojové prostředí, knihovnu [OpenCV](#) implementující algoritmy pro zpracování obrazu a počítačové vidění, a vlastní registrační framework, dlouhodobě vyvíjený na FIT VUT, přiložený ke zdrojovým kódům projektu ve složce 3rdparty.

Použití

Ukázky použití nástrojů v rámci případové studie jsou v balíčku k dispozici v rámci dávkových souborů.

Generování data setů

Vstupem pro generování data setů jsou trojrozměrné modely balení a produktu, které by měly být zarovnané k hlavním osám a rozměry by měly mít v metrovém měřítku, kalibrace kamery a limity pro generování náhodných poloh objektů.

Skripty pro generování data setů:

- **RandomPoses.py**

```
python RandomPoses.py <limitsFile> <numberOfSamples>
```

- <limitsFile> Název souboru obsahující minimální a maximální limity odchylek od referenční pózy
- <numberOfSamples> Počet náhodných poloh k vygenerování

Slouží ke generování náhodných odchylek od referenční pózy trojrozměrného modelu. Generované pózy jsou vypsány na standardní výstup.

- **Renders.py**

```
blender.exe --background --python Renders.py -- <cameraCalibration> <productModel>
<productReferencePose> <productRandomPoses> <packageModel> <packageReferencePose>
<packageRandomPoses> <outputDirectoryPath>
```

- <cameraCalibration> Cesta k souboru ve formátu CSV, který obsahuje údaje o kalibraci kamery; polohu kamery a ohniskovou vzdálnost v pořadí rx,ry,rz,tx,ty,tz,f.
- <productModel> Polygonální model produktu včetně textury ve formátu OBJ.
- <productReferencePose> Referenční póza modelu produktu.
- <productRandomPoses> Soubor ve formátu CSV obsahující odchylky od referenční pózy, vygenerovaný pomocí skriptu RandomPoses.py.
- <packageModel> Naskenovaný model balení ve formátu OBJ.
- <packageReferencePose> Referenční poloha balení.
- <packageRandomPoses> Odchylky od referenční pózy pro model balení.
- <outputDirectoryPath> Adresář, do kterého mají být ukládány renderované snímky.

Skript pro dávkové renderování data setu v modelovacím nástroji Blender. Na obrázku 2 jsou ukázky renderovaných snímků.



Obrázek 2: Ukázky renderovaných snímků z vytvořeného data setu.

- **MaskRenders.py**

```
blender.exe --background --python MaskRenders.py -- <cameraCalibration> <model> <referencePose>
<randomPoses> <outputDirectoryPath>
```

- <cameraCalibration> Soubor CSV s kalibrací kamery, ve stejném formátu jako u skriptu Renders.py.
- <model> Model (produktu, nebo balení) ve formátu OBJ.
- <referencePose> Referenční póza zadaného modelu.
- <randomPoses> Náhodné odchylky od referenční polohy modelu, pro které mají být masky vyrenderovány.
- <outputDirectoryPath> Adresář pro ukládání masek.

Nástroj ke generování masek objektů z vytvořeného data setu. Maskování je řešeno pomocí alfa kanálu.

- **RenderRotations.py**

```
blender.exe --background --python RenderRotations.py -- <cameraCalibration> <model> <referencePose> <offset> <outputDirectoryPath>
```

- <cameraCalibration> Kalibrace kamery.
- <model> Model produktu.
- <referencePose> Referenční póza modelu.
- <offset> Posunutí na podélné ose modelu.
- <outputDirectoryPath> Adresář pro ukládání výsledných snímků.

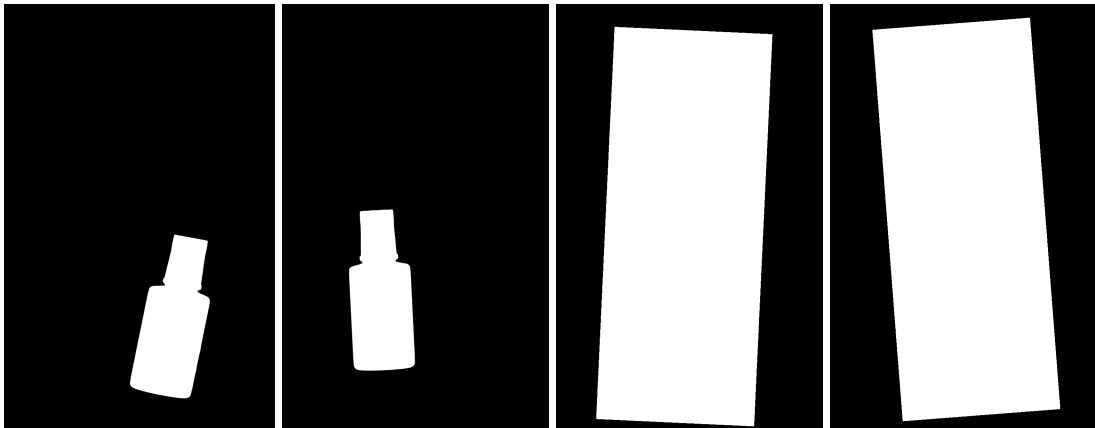
Skript sloužící k vytvoření souboru dat pro trénování regresní konvoluční sítě, určené k odhadu rotace symetrického objektu kolem jeho podélné osy. Skript vygeneruje 360 obrázků, které se liší natočením objektu po 1°. Posunutí menší než jeden stupeň lze nastavit parametrem offset.

- **BinarizeMasks.py**

```
python BinarizeMasks.py <inputDirectoryPath> <outputDirectoryPath>
```

- <inputDirectoryPath> Adresář obsahující vyrenderované masky s alfa kanálem.
- <outputDirectoryPath> Adresář pro ukládání výsledných binárních masek.

Převod masek z barevného snímku s alfa kanálem do binární podoby, ukázka je k dispozici na obrázku 3.



Obrázek 3: Binarizované rastrové masky.

- **Masks2Coco.py**


```
python Masks2Coco.py <inputImages> <productBinaryMasks> <packageBinaryMasks> <rangeFrom> <rangeTo> <outputJSONFile>
```

- <inputImages> Původní snímky z data setu.
- <productBinaryMasks> Binární masky označující produkt v původních snímcích.
- <packageBinaryMasks> Binární masky označující balení v původních snímcích.
- <rangeFrom> Číslo prvního snímku, který má být zahrnut do data setu.
- <rangeTo> Číslo posledního snímku, který má být přítomný v data setu.
- <outputJSONFile> Jméno pro výstupní JSON soubor ve formátu COCO.

Provádí vektorizaci binárních masek a vytváří JSON soubor ve formátu COCO, popisující vytvořený data set. Metadata data setu se editují přímo v souboru skriptu.

- **Coco2Detectron2.py**

```
python Coco2Detectron2.py <inputCocoJSON> <outputDetectron2JSON>
```

- <inputCocoJSON> Soubor ve formátu COCO určený k převedení do formátu Detectron2.
- <outputDetectron2JSON> Jméno pro výstupní soubor ve formátu pro Detectron2.

Provádí převod z formátu COCO do formátu pro knihovnu Detectron2. Na obrázku 4 je ukázka vizualizace náhodných snímků včetně segmentací z výsledného data setu.



Obrázek 4: Vizualizace snímků z data setu převedeného do formátu podporovaného knihovnou Detectron2. Ilustrace byly vytvořeny pomocí nástroje Visualizer z balíčku detectron2.utils.visualizer.

Segmentace

Segmentace se provádí pomocí modelu konvoluční neuronové sítě zvaného Mask-RCNN. Software obsahuje nástroje pro trénování neuronové sítě a pro predikci segmentace na nových nebo testovacích případech.

Obsažené skripty:

- **TrainMaskRCNN.py**

```
python TrainMaskRCNN.py <trainDataSetJSON> <imagesPath> <outputFileName>
```

- <trainDataSetJSON> JSON soubor popisující data set ve formátu Detectron2.
- <imagesPath> Cesta ke snímkům data setu.
- <outputFileName> Jméno souboru, pod kterým se má uložit výsledná natrénovaná neuronová síť.

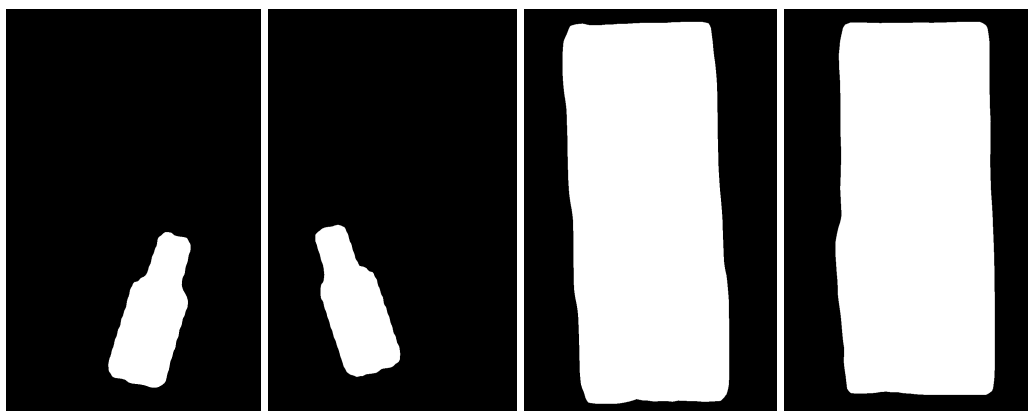
Provádí trénování konvoluční neuronové sítě na vytvořeném data setu.

- **PredictMaskRCNN.py**

```
python PredictMaskRCNN.py <maskRCNNfile> <inputImage> <outputProductMaskDir>  
<outputPackageMaskDir>
```

- <maskRCNNfile> Soubor, ve kterém je uložená natrénovaná síť Mask-RCNN..
- <inputImage> Cesta ke vstupnímu obrázku, na kterém má být provedena segmentace.
- <outputProductMaskDir> Složka, do které se má uložit výstupní binární maska pro produkt.
- <outputPackageMaskDir> Umístění, kam se má uložit binární maska pro balení.

Slouží k predikci segmentace na nových snímcích. Ukázky výsledných binárních masek jsou na obrázku 5.



Obrázek 5: Binární masky získané segmentací testovacích snímků.

Rekonstrukce trojrozměrné polohy

Slouží k odhadu vlastní polohy v prostoru daného produktu a balení na základě získaných binárních masek a kalibrace kamery. Modely pro 2D-3D registraci vyžadují oproti modelům pro 2D-3D registraci milimetrové měřítko.

- **RegisterModel.exe**

```
registerModel.exe <inputMask> <model> <cameraCalibration> <startingPose> <outputImage>
<resultPose> <iterationImages> <verbose>
```

- <inputMask> Vstupní maska získaná ze segmentace neuronovou sítí.
- <model> Cesta k modelu, který má být do masky registrován. Model musí být v milimetrovém měřítku.
- <cameraCalibration> CSV soubor s kalibrací kamery.
- <startingPose> Startovací póza pro optimalizaci polohy.
- <outputImage> Výstupní obrázek produktu ve standardizované poloze, zbavený pozadí a změněný na velikost 256 x 256 pixelů. Pokud je parametr nastaven na "none", poloha se neukládá.
- <resultPose> CSV s výslednou 3D pózou modelu.
- <iterationImages> Adresář, do kterého se ukládají snímky po každé iteraci. Pokud je parametr nastaven na "none", snímky se neukládají.
- <verbose> Pokud je parametr nastaven na 1, zobrazují se informace o objektivní funkci během každé iterace.

Provádí 2D-3D registraci modelu do vysegmentované binární masky. Výstupem je jak odhad polohy v prostoru, tak standardizované snímky produktu pro další zpracování, resp. odhad rotace kolem podélné osy objektu. Ukázky standardizovaných snímků po registraci jsou na obrázku 6.



Obrázek 6: Standardizované snímky produktu jako jeden z výstupů 2D-3D registrace.

Zpřesnění odhadu polohy

Odhad rotace symetrického produktu kolem jeho podélné osy se provádí pomocí regresní konvoluční sítě.

- **TrainRotationCNN.py**

```
python TrainRotationCNN.py <rotationDataSet1> <rotationDataSet2> <boundingBox> <outputCNNFile>
```

- <rotationDataSet1> Adresář k první části trénovacího data setu.
- <rotationDataSet2> Složka s druhou částí trénovacího data setu.
- <boundingBox> Soubor ve formátu CSV obsahující bounding box produktu na snímcích trénovacího data setu.
- <outputCNNFile> Soubor, do kterého se má uložit výsledná regresní síť.

Nástroj pro trénování konvoluční sítě pro odhad rotace kolem podélné osy objektu.

- **PredictRotationCNN.py**

```
python PredictRotationCNN.py <rotationCNNfile> <inputImage> <outputFile>
```

- <rotationCNNfile> Soubor, ve kterém je uložena natrénovaná síť pro odhad rotace objektu.
- <inputImage> Cesta ke vstupnímu obrázku, podle kterého se má provést predikce.
- <outputFile> Soubor, do kterého se má uložit predikovaný úhel rotace objektu.

Nástroj pro predikci rotace objektu okolo podélné osy. Vstupní snímek musí mít standardizovanou podobu, velikost snímku musí být 256 x 256 pixelů.

Reference

- [Klima2016] KLÍMA Ondřej, KLEPÁRNÍK Petr, ŠPANĚL Michal a ZEMČÍK Pavel. Intensity-based femoral atlas 2D/3D registration using Levenberg-Marquardt optimisation. In: Biomedical Applications in Molecular, Structural, and Functional Imaging. San Diego: SPIE - the international society for optics and photonics, 2016, s. 1-6. ISBN 978-1-5106-0023-2.
- [He2018] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, June 2018.

Příloha A

OPEN SOURCE LICENCE VUT V BRNĚ

Verze 1.

Copyright (c) 2010, Vysoké učení technické v Brně, Antonínská 548/1, PSČ 601 90

INSTALACÍ, KOPÍROVÁNÍM NEBO JINÝM POUŽITÍM SOFTWARE POTVRZUJETE, ŽE SOUHLASÍTE S PODMÍNKAMI TÉTO LICENČNÍ SMLOUVY. POKUD S TĚMITO PODMÍNKAMI NESOUHLASÍTE, SOFTWARE NEINSTALUJTE, NEKOPÍRUJTE ANI NEPOUŽÍVEJTE.

JESTLIŽE NEMÁTE PLATNOU LICENCI NA SOFTWARE, NEJSTE OPRÁVNĚNI SOFTWARE INSTALOVAT, KOPÍROVAT ANI JINAK POUŽÍVAT.

Definice:

Software se pro účely této smlouvy rozumí počítačový program (skupina počítačových programů tvořící jeden funkční celek) schopný autorskoprávní ochrany a s ním související dokumentace.

Dílem založeným na Software se pro účely této smlouvy rozumí dílo, které obsahuje zcela nebo jen zčásti Software nebo jakoukoli jeho část, popřípadě je ze Software nebo jeho části odvozeno. Na části díla schopného samostatného užití, které neobsahují žádnou část Software ani nejsou ze Software odvozeny, se tato definice nevztahuje, pokud jsou šířeny samostatně.

Zdrojovým kódem se pro účely této licence rozumí veškerý zdrojový kód pro všechny moduly, které Software nebo dílo na něm založené obsahuje, plus jakékoli další soubory pro definici rozhraní, plus dávkové soubory potřebné pro kompilaci a instalaci spustitelného programu. Zvláštní výjimkou jsou však ty programové komponenty, které jsou normálně šířeny (buď ve zdrojové, nebo binární formě) s hlavními součástmi operačního systému, na němž spustitelný program běží (tj. s překladačem, jádrem apod.).

Každý, kdo použije Software, stává se **Uživatelem**. Uživatel se zavazuje dodržovat tyto licenční podmínky.

VUT V BRNĚ UDĚLUJE UŽIVATELI LICENCI K UŽÍVÁNÍ SOFTWARE ZA NÁSLEDUJÍCÍCH PODMÍNEK:

1. Uživatel smí užívat Software zdarma pro vlastní potřebu k jakýmkoliv účelům.
2. Uživatel smí kopírovat a šířit doslovné kopie spustitelného Software a zdrojového kódu Software tak, jak jej obdržel, na libovolném médiu, za předpokladu, že na každé kopii viditelně a náležitě zveřejní zmínku o autorských právech a absenci záruky, ponechá nedotčené všechny zmínky vztahující se k této licenci a k absenci záruky a dá každému příjemci spolu se Software kopií této licenční smlouvy. Za fyzický akt přenesení kopie může Uživatel žádat poplatek a podle vlastního uvážení může nabídnout za poplatek záruční ochranu.
3. Uživatel může modifikovat svoji kopii či kopie Software anebo kterékoliv jeho části, a tak vytvořit dílo založené na Software, a kopírovat a rozšiřovat takové modifikace či dílo, pokud jasně uvede, že se jedná o modifikované dílo. Tyto modifikace či dílo lze licencovat pouze za podmínek čl. 2 těchto licenčních podmínek, bez ohledu na to, zda jsou šířeny samostatně nebo ve spojení s jiným dílem.
Ustanovení předchozí věty se však nepoužije na díla, která jsou pouhým spojením jiného díla, jež není na Software založeno, se Software nebo dílem založeným na Software na paměťovém nebo distribučním médiu.
4. Kopie Software nebo díla na něm založeného v objektové anebo spustitelné podobě musí Uživatel:
 - a) doprovodit zdrojovým kódem ve strojově čitelné formě. Zdrojový kód musí být rozšiřován podle ustanovení čl. 2, a to na médiu běžně používaném pro výměnu programového vybavení; nebo
 - b) doprovodit písemnou nabídkou nejméně na tři roky, podle níž poskytne Uživatel jakékoli třetí straně, za poplatek nepřevyšující výdaje skutečně vynaložené na fyzickou výrobu zdrojové distribuce, kompletní strojově čitelnou kopii odpovídající zdrojovému kódu, jenž musí být šířen podle ustanovení čl. 2 na médiu běžně používaném pro výměnu programového vybavení; nebo
 - c) doprovodit informacemi, které dostal ohledně nabídky na poskytnutí zdrojového kódu. (Tato alternativa je povolena jen pro nekomerční šíření a jenom tehdy, pokud Uživatel obdržel program v objektovém nebo spustitelném tvaru spolu s takovou nabídkou, v souladu s písmenem b výše.)
5. Uživatel nesmí kopírovat, modifikovat, poskytovat sublicence anebo šířit Software jiným způsobem než výslovně uvedeným v této licenci. Jakýkoli jiný pokus o kopírování, modifikování, poskytnutí sublicence anebo šíření Software je neplatný a automaticky ukončí práva daná touto licenci. Třetí osoby, které od Uživatele obdržely kopie anebo práva v souladu s touto licenci, však nemají své sublicence ukončeny, dokud je jejich chování v souladu s těmito licenčními podmínkami.

6. Pokud nemůže Uživatel šířit Software tak, aby vyhověl zároveň svým závazkům vyplývajícím z této smlouvy a jiným platným závazkům, nesmí jej v důsledku toho šířit vůbec.
7. Uživatel není odpovědný za vymáhání dodržování podmínek této licence třetími stranami.
8. VZHLEDEM K BEZPLATNÉMU POSKYTNUTÍ LICENCE K SOFTWARE SE NA SOFTWARE NEVZTAHUJE ŽÁDNÁ ZÁRUKA, A TO V MÍŘE POVOLENÉ ZÁKONEM. POKUD NENÍ PÍSEMNĚ STANOVENO JINAK, POSKYTUJE VUT V BRNĚ SOFTWARE "TAK, JAK JE", BEZ ZÁRUKY JAKÉHOKOLIV DRUHU, AŽ VÝSLOVNÉ NEBO KONKLUDENTNÍ, VČETNĚ, ALE NEJENOM, ZÁRUK ZA KVALITU A VÝKONNOST SOFTWARE, ZA PRODEJNOST A VHODNOSTI PRO URČITÝ ÚČEL.
VUT V BRNĚ V ŽÁDNÉM PŘÍPADĚ NEODPOVÍDÁ ZA ŠKODY, VČETNĚ VŠECH OBECNÝCH, SPECIÁLNÍCH, SKUTEČNÝCH NEBO UŠLÉHO ZISKU VYPLÝVAJÍCÍCH Z UŽÍVÁNÍ ANEBO NESCHOPNOSTI UŽÍVAT SOFTWARE (VČETNĚ ALE NIKOLI JEN, ZTRÁTY NEBO ZKRESLENÍ DAT, NEBO TRVALÝCH ŠKOD ZPŮSOBENÝCH VÁM NEBO TŘETÍM STRANÁM, SELHÁNÍ FUNKCE SOFTWARE V SOUČINNOSTI S JINÝMI SOFTWARE, NEBO V PŘÍPADĚ, ŽE PROGRAM PORUŠUJE PRÁVA TŘETÍCH OSOB), A TO I V PŘÍPADĚ, ŽE VUT V BRNĚ VĚDELO O MOŽNOSTI TAKOVÝCH ŠKOD, TO VŠE V NEJŠIRŠÍ MOŽNÉ MÍŘE POVOLENÉ ZÁKONEM.

Závěrečná ustanovení:

Nevynutitelnost anebo neplatnost anebo neúčinnost kteréhokoli ujednání této smlouvy neovlivní vynutitelnost anebo platnost anebo účinnost jejích ostatních ustanovení.

Tyto licenční podmínky poskytují stejný rozsah licenčních oprávnění jako licenční podmínky GNU GPL licencí a Software splňuje podmínky pro označení Open Source Software.

Otázky touto smlouvou neupravené včetně otázek jejího vzniku a zániku se řídí českým právem.

V případě soudního sporu budou příslušné české obecné soudy.

Instalací, kopírováním nebo jiným použitím Software Uživatel prohlašuje, že se s obsahem této smlouvy důkladně seznámil, že je souhlas s touto smlouvou projevem jeho skutečné, vážné, svobodné a určité vůle prosté omylu a že není uzavřena v tísní za nápadně nevýhodných podmínek.