

VYSOKÉ UČENÍ TECHNICKÉ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Biometrické systémy Generování syntetického otisku prstu z biometrické šablony

1 Zadání

Vytvořte aplikaci, která vytvoří syntetický otisk prstu na základě reálné šablony. Cílem je tedy otisk, který vypadá jako otisk a obsahuje minimálně všechny markanty ze šablony (může obsahovat i nějaké navíc) + odpovídá třída otisku. K dispozici jsou existující generátor otisků a komerční nástroje pro případnou tvorbu šablon a vyhodnocení.

Covid rozšíření: Budou provedeny další zpřesňující (automatizované) nastavení generování a možnost načtení profilu generování ze souboru. Automatizované nastavení bude odhadnuto z informací obsažených v šabloně, načtení profilu pak bude možno provést (importovat i exportovat) manuálně. Nastaveními jsou: nastavení třídy otisku (počet tříd případně rozšiřte), nastavení hustoty v otisku, nastavení velikosti otisku (tvar otisku) a šířky linií.

2 Úvod

Naším úkolem bylo vytvořit aplikaci, která vytvoří syntetický otisk prstu na základě reálné šablony. Tento otisk má obsahovat všechny markanty ze šablony a odpovídá mu třída otisku prstu. Měli jsme k dispozici existující program **SyFDaS**, který již v sobě měl řadu funkcí implementovanou.

Nejprve se v sekci 3 podíváme na existující generátory (včetně SyFDaS) a popíšeme, jak generování probíhá. Následně v sekci 4 popíšeme existující šablony pro uchovávání informací o otisku, ze kterých jsme čerpali, a nakonec v sekci 5 se zaměříme konkrétně na implementované části do programu SyFDaS.

3 Existující generátory otisků prstů

Jelikož se biometrické systémy stále rozšiřují, je potřeba klást větší důraz na kvalitu rozpoznávacích algoritmů. V tomto nám mohou pomoci generátory syntetických otisků prstů, díky kterým si můžeme jednoduše a levně vytvořit testovací databázi, na základě které se mohou rozpoznávací algoritmy učit. Při rešerši existujících programů na generování jsme jako první narazili na program **SFinGe**, který nás hned oslovil. Umožňuje totiž velmi detailní nastavení výsledného otisku a zdál se nám velmi intuitivní.

Generování probíhá na následujícím principu. Nejprve je vygenerován tvar otisku, směrová mapa papilárních linií a mapa hustoty otisku. Vše je vygenerováno samostatně. Otisk se vytvoří metodou prostorového lineárního filtrování, kde orientace a četnosti filtrů jsou lokálně upraveny podle předešlé vygenerované mapy a hustoty. V dalším kroku se přidá šum a pozadí pro přidání realismu. Takto vytvořený otisk se pak použije pro generování podobných otisků, které mají stejné znaky, ale liší se tvarem. [3], [1]

3.1 SyFDaS

SyFDaS je generátor otisků prstů a simulátor poškození, vytvořený panem Ing. Ondřejem Kanichem ve spolupráci s výzkumnou skupinou STRaDe. Vychází z metody SFinGe a generuje otisky na základě znaků neboli markantů. Těchto znaků je několik druhů, liší se nejen geometrickým tvarem, ale i četností v lidské populaci. Základními typy jsou: začátek/ukončení linie a jednoduchá vidlice, ze kterých se dají vytvořit všechny ostatní odvozené markantní body. Na obrázku 1 lze vidět vybrané typy markantů.

U takto nalezených markantních bodů je pak pozornost hlavně směřována k jejich pozici a směrové orientaci v prostoru.



Obrázek 1: Vybrané typy markantů

4 Popis šablony

V rámci projektu jsme neměli přesně určenou šablonu, kterou bychom měli použít ke generování. Vybrali jsme si proto 4 šablony, jejichž popis jsme našli na stránkách <https://templates.machinezoo.com>. Jedná se o následující šablony.

- ANSI INCITS 378-2004
- ANSI INCITS 378-2009[R2014]
- ISO/IEC 19794-2:2005
- ISO/IEC 19794-2:2011

Tyto formáty byly zveřejněny ANSI a ISO, a jsou zde o nich uvedeny dostatečné informace bez nutnosti kupovat si originální specifikaci. Autorem dokumentu je Robert Važan¹, který popis zveřejnil jako součást open source knihoven SourceAFIS a FingerprintIO a on sám nemá žádné spojení s ISO/ANSI.

Šablony jsou si velmi podobné, proto popíšeme detailněji pouze jednu. Ostatní obsahují stejná data, ale pořadí se může lišit. Vezměme si například šablonu *ANSI INCITS 378-2004*. Je složena z následujících bloků:

1. Hlavička (template header)
2. Otisk prstu (fingerprint)
3. Markanty (minutia)
4. Ostatní data (extension data block)

Pokud se nachází v extension bloku typ 1, pak následuje blok pro **ridge count** (počet papilárních linií). Pokud je zde uvedena hodnota 2, datový blok obsahuje informace o obsažených corech a deltách. Podrobná struktura šablony ANSI 2004 je vidět na obrázku 2.

Pro potřeby generování budeme hlavně používat tato pole (duplikované informace se odvíjí od verze standardu) :

- **hlavička** - výška, šířka obrázku, hustota, počet otisků,
- **otisky prstů** - výška, šířka obrázku, počet markantů,
- **markanty** - typ, pozice, úhel,
- **extension data blok** – informace o corech a deltách (pozice, úhel, počet).

Jak je uvedeno na stránkách <https://templates.machinezoo.com>, většina systémů pro rozpoznávání otisků používá vlastní šablonu. NIST navíc používá formát *CBEFF* (Common Biometric Exchange File Format²), který je jakýmsi dalším zaobalením těchto standardů. Přesnou specifikaci formátu CBEFF, který odpovídá databázi šablon od NISTu jsme však nemohli dohledat, proto jsme se rozhodli si šablony vytvářet sami a formát CBEFF tedy nebereme v úvahu.

5 Implementace

V rámci implementace jsme v projektu SyFDaS upravovali pouze obsah složky *FpGenerator*.

¹<https://robert.machinezoo.com/>

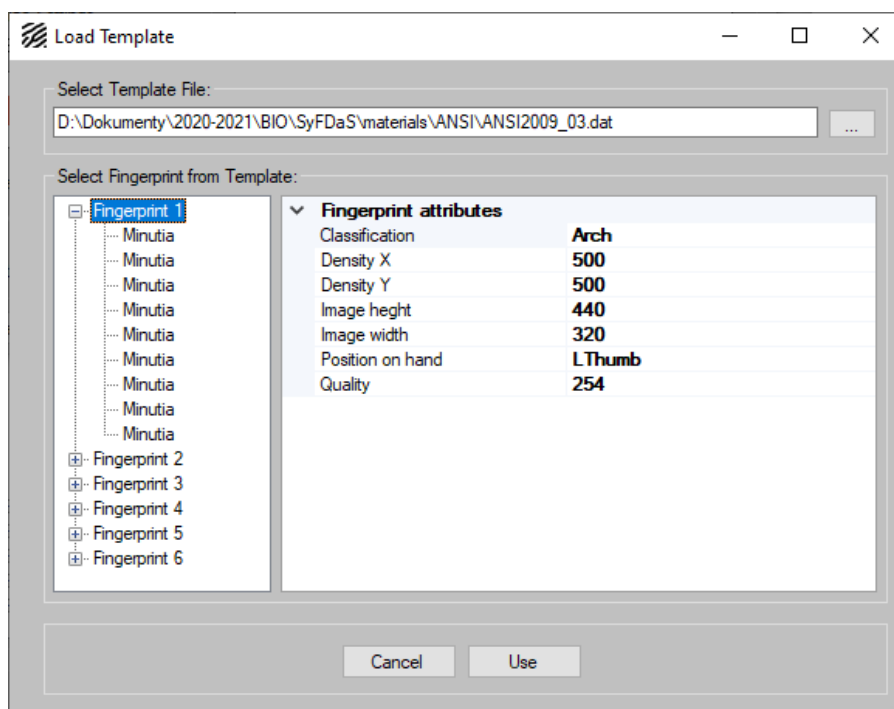
²<https://www.nist.gov/publications/common-biometric-exchange-file-format-cbeff>

5.1 Generování ze šablony

Celý proces generování šablony je zamýšlen v tomto pořadí:

1. Uživatel vybere možnost generování ze šablony.
2. V novém okně dostane možnost zvolení souboru se šablonou.
3. Z nahrané šablony dostane na výběr konkrétní otisk prstu.
4. Po zvolení otisku prstu se do programu automaticky vygenerují další upřesňující informace, které si uživatel může pozměnit.
5. Uživatel generuje otisk.

Okno pro výběr konkrétního otisku ze šablony lze vidět na obrázku 3.



Obrázek 3: Okno pro výběr konkrétního otisku prstu z načtené šablony

Hlavní funkcionalita parsování šablony byla naimplementována do třídy `FpTemplate`. Posloupnost událostí při generování byla implementována do třídy `FpGeneratorPresenter` do metody, která obsluhuje možnost načtení otisku ze šablony.

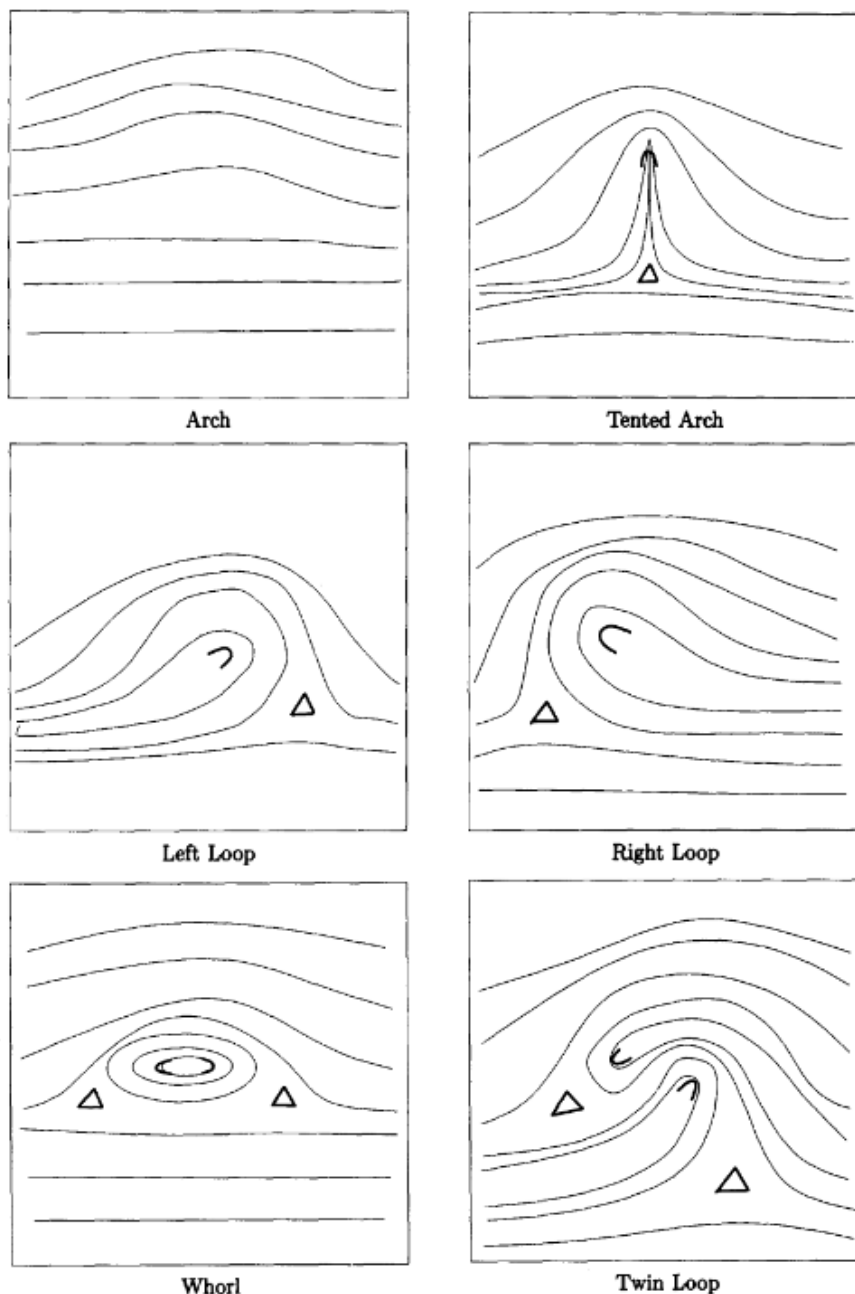
5.2 Nahrávání šablony

Prvním krokem je rozpoznání, o který typ šablony se jedná. Jelikož oba standardy ISO i ANSI používají stejné hodnoty polí, která je mají rozeznávat, tak rozeznání probíhá nejprve na základě hodnoty *VERSION* v hlavičce, tím zjistíme jestli se jedná o staré nebo nové verze standardů ISO/ANSI. A dále pak rozhodujeme na základě pole *TOTALBYTES*. O konkrétním postupu píše Robert Važan stránce <https://templates.machinezoo.com>, kdy rozeznáváme první a druhý bajt tohoto pole a rozeznání probíhá na základě úvahy o reálných hodnotách tohoto pole (pokud je číslo až příliš velké pro jeden styl parsování a normální pro druhé, tak rozhodneme že se jedná o druhý typ)

Dále je parsování už více přímočaré. Postupně se prochází otisky v šabloně a jejich markanty a další údaje se předělávají na vnitřní strukturu. Uživatel díky tomu dostane na výběr, který otisk z šablony chce generovat a může i změnit atributy tohoto otisku (viz obrázek 3).

5.3 Klasifikace třídy otisku

Naše implementace dokáže rozeznat 5 tříd otisku prstu - arch, tented arch, right loop, left loop a whorl. Jediná chybějící třída je twin loop, které chyběla v SyFDaSu. Bohužel jsme se nedostali k jejímu přidání. Při rozpoznání třídy otisku jsme čerpali z materiálu Fingerprint Classification od Kalle Karu a Anil K. Jain [4].



Obrázek 4: Cores a deltas pro různé třídy otisku

Třídy otisků prstu lze rozpoznat na základě útvarů zvaných **core** a **delta**. Core je střed otisku, kolem kterého se obtáčí papilární linie. Delta je jakýsi "trojúhelníček", kde se linie rozcházejí ve třech směrech.

Prvním znakem, na základě čeho lze rozpoznat třídu, je počet výše zmíněných útvarů. V případě třídy arch není přítomna ani jedna core ani delta, proto už můžeme jednoduše rozlišit tuto třídu. Poté nastávají situace, že je v otisku právě jeden core a právě jedna delta nebo právě dva cores a právě dvě delty. V případě jednoho core a jedné delty se nám nabízí tři třídy. Jako první krok tedy zjistíme sklon

přímky, kterou vytvoříme mezi corem a deltou. Pokud se core a delta nachází pod sebou, tedy sklon je kolem 90 stupňů, jedná se o tended arch. Pokud ne, zjistíme jen jestli je delta napravo nebo nalevo (znaménko sklonu) a určíme jestli se jedná o right loop nebo left loop.

V případě dvou cores a dvou deltas jsme v naší implementaci rozhodli o klasifikaci na whorl, kvůli chybějící třídě twin loop. Zde by se musela spočítat vzájemná pozice cores a z ní usoudit třídu otisku.

5.4 Rozpoznání šířky linií

Jelikož šablona neobsahuje žádné konkrétní informace ohledně šířky papilárních linií, tak jsme museli tuto šířku odhadnout na základě markantů. Standardně se šířka linie pohybuje kolem 0.2 – 0.7 mm [5]. Pokud se tedy nějaké dva markanty nachází nad sebou, budou právě v tomto rozmezí.

Na základě této úvahy tedy nejprve najdeme dva nejbližší markanty. Porovnáme, zda jejich vzdálenost leží v definovaném rozmezí, které odpovídá standardní šířce linií. Pokud ano, přepočítáme vzdálenost na hodnotu lambdy pro Gaborův filtr (délka vlny).

Přepočítání je na základě transformace hodnoty jednoho intervalu (minimální vzdálenost a standardní šířka linie) na hodnotu druhého intervalu (rozmezí hodnot pro lambda). Rozmezí hodnot pro lambda jsme získávali vizuálním posudkem, kdy jsme generovali různé otisky z různými hodnotami. Aktuálně je rozmezí nastaveno na interval 4.5 – 10.

5.5 Vycentrování otisku

Jelikož se v SyFDaSu nachází cores a deltas na předem určených místech, je třeba před samotným generováním posunout všechny markanty tak, aby jejich pozice vůči core/delta byla přibližně stejná i ve vygenerovaném otisku. Ve třídě arch toto není potřeba řešit, jelikož nemáme k dispozici ani core, ani deltu. Ve třídě tended arch centrujeme pomocí delty, tedy vypočítáme její ofset od středu a všechny markanty posuneme o tento ofset. Ve třídě whorl vypočítáme střední bod mezi cores a poté ofset tohoto bodu od středu, pomocí kterého vycentrujeme markanty. V ostatních třídách vycentrujeme pomocí offsetu core od středu.

5.6 Generování masky

Masku otisku lze odhadnout na základě pozic markantů – ty nám pomohou stanovit krajní hranice masky. Déle bude potřeba otisk upravit tak, aby vypadal jako otisk. Proto je potřeba mu zaoblit rohy a zúžit lehce vrchní část oproti spodní. Experimentálně nám dával nejlepší výsledky vzorec uvedený v metodě *GenerateMaskFromTemplate* v *FpGeneratorPresenter*. Postup je uvedený v následujícím seznamu.

1. Nalezení minimální a maximální hodnoty ve směru x,y podle markantů
2. Přidání okraje
3. Zúžení vršku otisku podle šířky otisku
4. Prodloužení otisku podle výšky celkového otisku

5.7 Načítání a ukládání profilu

V rámci rozšíření jsme měli načítat a ukládat nastavení profilu otisku. Přidali jsme tuto možnost do nového tabu *Options*. Profil ukládáme ve formátu json. Ukládaný soubor vypadá následovně:

```
{
  "mask": {
    "MaskWidth" :Int
    "MaskHeight" :Int
    "AllPoints" :[Point]
  },
}
```

```

"orientation": {
    "OrientationClass": Int,
    "MinutiaPower": Int
},
"density": {
    "BrushSize": Int,
    "ChangeType": 1|0,
    "Alpha": Int,
    "DensityBitmapInBytes" : []
},
"filter": {
    "Gamma": Float,
    "Delta": Float,
    "Theta": Int,
    "Lambda": Float,
    "Sigma": Float,
}
}

```

Třída obsahující práci s profilem se nazývá *FpProfile* a najdeme ji ve složce *Model*. Obsahuje metodu *ProfileFromFile*, která rozparsuje json na vstupu a uloží získané hodnoty. Ukládání probíhá v *FpGeneratorPresenter* (metoda *SaveProfile*), který vytvoří novou instanci *FpProfile* a zapíše výsledný json pro uložení.

6 Testování a ověření

Jak již bylo zmíněno, databáze od NISTu jsme nemohli použít kvůli absenci specifikace formátu CBEFF, který tato data používají. Proto jsme se rozhodli si vygenerovat testovací šablony sami. Zde je uvedený postup:

1. Nejdříve jsme si pomocí generátoru SFinGe vygenerovali otisk prstu, ve kterém orientační mapa je přibližně stejná jako ta v SyFDaSu (Obrázek 5 vpravo).
2. Pomocí programu Fingerprint Minutiae Viewer Beta v3.0 od NISTu jsme si zobrazili markanty v otisku prstu (Obrázek 5 vlevo).
3. Jelikož, předchozí software neumí exportovat markanty v šablonách, použili jsme software of Neurotec Biometric pro vytvoření šablony ručně. V ukázkové aplikaci FMRecord Editor lze vytvářet a editovat šablony ve formátech, které jsme se rozhodli podporovat.
4. Uloženou šablonu jsme nahráli do SyFDaSu a upravili jsme masku, aby odpovídala původnímu otisku. Vygenerovali jsme otisk (Obrázek 6).
5. Posoudili jsme, zda vygenerovaný otisk vypadá podobně jako původní otisk. tedy zda-li odpovídá jeho třída a zda-li se markanty nachází přibližně na místech, na kterých mají být.



Obrázek 5: Původní otisk prstu, ze kterého vznikla šablona



Obrázek 6: Vygenerovaný otisk prstu ze šablony

7 Nalezené problémy a možnosti vylepšení

Jelikož program SyFDaS pracuje s již předem vytvořenými orientačními poli a ne s cores a deltas, tak je i naše řešení omezeno na otisky, jejichž orientační pole je podobné těm ze SyFDaSu. Řešením by nejspíše bylo nastavování cores a deltas v orientačním poli, což jsme již neimplementovali.

Dalším omezením je již zmíněná chybějící třída twin loop. Pokud by se přidala (respektive její orientační pole), musel by se rozšířit klasifikátor třídy šablony.

Během implementace jsme narazili na pár nedostatků SyFDaSu, které končily výjimkami. Snažili jsme se tyto nedostatky opravit, což se nám z větší části povedlo. Největšími nedostatky byly:

- **New Fingerprint nabídka v menu** – po vybrání této možnosti se ukázal dialog, který když se beze změny odsouhlasil tak vyvolal výjimku. Problém byl v tom, že defaultní hodnoty v dialogu nebyly nastaveny i v programu a program se snažil vytvořit obrázek 0x0.

- **Reset prostředí** – pokud byly v prostředí vybrány možnosti jiné než defaultní (zaškrtnuté checkboxy), tak neproběhnul správný restart prostředí při novém obrázku. Na vině byly události, které se volaly při pokusu o restart prostředí z kódu.

8 Závěr

V rámci projektu se nám nakonec podařilo rozšířit existující generátor SyFDaS o funkce: generování ze šablony, ukládání a načítání profilu vygenerovaného otisku prstu. Podporujeme 4 typy šablon, podle kterých do programu vstupují informace o obrázku, dále počet a pozice markantů, které slouží k rozpoznání třídy otisku, vygenerování masky a odhadu šířky linií. Implementovaná část programu byla otestována pomocí různých existujících programů pro vytváření šablon a pro generování otisku.

Reference

- [1] WILDMANN, Radek. *Generování projevu kožního onemocnění do syntetických otisků prstůz generátoru SFinGe*. Brno, 2019. *Bakalářská práce*. Vysoké učení technické v Brně, Fakulta-informačních technologií. Vedoucí práce Ing. Ondřej Kanich, Online: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=198222
- [2] Ondřej Kanich: *Research in Fingerprint Damage Simulations*, doctoral thesis, Brno, FIT BUT, Brno, Czech Republic, 2018, Online: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=206270
- [3] SMÉKAL, O. *Biometrie otisku prstu: Bakalářská práce*. Brno: FEKT VUT v Brně, 2010, 42s., 2 příl. Vedoucí bakalářské práce Ing. Petr Fedra, Online: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=28074
- [4] Kalle Karu, Anil K. Jain, *Fingerprint classification*. Volume 29, Issue 3, 1996, ISSN 0031-3203
- [5] *Papilární linie*. Poslední aktualizace: 5. 10. 2017, Online: https://cs.wikipedia.org/wiki/Papil%C3%A1rn%C3%AD_linie

	26+ bytes		ANSI INCITS 378-2004 template
once	25 or 29 bytes	HEADER	Template header
once	4 bytes	MAGIC	File signature / magic number
once	4 bytes	VERSION	Format version
once	2 or 6 bytes	TOTALBYTES	Total template length in bytes
once	2 bytes	VENDOR	Vendor ID
once	2 bytes	SUBFORMAT	Vendor-specified subformat
once	4 bits	DEVSTAMP	Sensor compliance
once	12 bits	DEVID	Sensor ID
once	2 bytes	WIDTH	Image width
once	2 bytes	HEIGHT	Image height
once	2 bytes	RESOLUTIONX	Horizontal pixel density
once	2 bytes	RESOLUTIONY	Vertical pixel density
once	1 byte	FPCOUNT	Number of fingerprints
once	1 byte		Reserved byte (zeroed)
FPCOUNT	6+ bytes	FINGERPRINT	Fingerprint
once	1 byte	POSITION	Finger position on hands
once	4 bits	VIEWOFFSET	Finger view number
once	4 bits	MEDIATYPE	Sensor category
once	1 byte	FPQUALITY	Fingerprint quality
once	1 byte	MINCOUNT	Number of minutiae
MINCOUNT	6 bytes	MINUTIA	Minutia
once	2 bits	MINTYPE	Minutia type
once	14 bits	MINX	Minutia X position
once	2 bits		Reserved (zeroed)
once	14 bits	MINY	Minutia Y position
once	1 byte	MINANGLE	Minutia angle
once	1 byte	MINQUALITY	Minutia quality
once	2 bytes	EXTBYTES	Total length of extension data
0 or more	4+ bytes	EXTENSION	Extension data block
once	2 bytes	EXTTYPE	Extension type
once	2 bytes	EXTLEN	Length of extension data
once	EXTLEN - 4	EXTDATA	Extension data

Obrázek 2: Popis šablony ANSI INCITS 378-2004 podle <https://templates.machinezoo.com>