



MATEMATICKO-FYZIKÁLNÍ FAKULTA Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Viktor Němeček

Percepční učení a Ideální Bayesovský pozorovatel při zrakovém vyhledávání

Katedra softwaru a výuky informatiky

Vedoucí bakalářské práce: Mgr. Filip Děchtěrenko, Ph.D.

Studijní program: Informatika

Studijní obor: Obecná informatika

Praha 2018

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Poděkování.

Název práce: Percepční učení a Ideální Bayesovský pozorovatel při zrakovém vyhledávání

Autor: Viktor Němeček

Katedra: Katedra softwaru a výuky informatiky

Vedoucí bakalářské práce: Mgr. Filip Děchtěrenko, Ph.D., Katedra softwaru a výuky informatiky

Abstrakt: Abstrakt.

Klíčová slova: klíčová slova

Title: Perceptual learning and Ideal Bayesian observer in visual search task

Author: Viktor Němeček

Department: Department of Software and Computer Science Education

Supervisor: Mgr. Filip Děchtěrenko, Ph.D., Department of Software and Computer Science Education

Abstract: Abstract.

Keywords: key words

Obsah

Úvod	2
Struktura práce	3
1 Teoretický úvod	4
1.1 Zrakové vyhledávání	4
1.2 Úvod do teorie detekce signálu	4
1.2.1 Senzitivita	5
1.3 Šum	7
1.3.1 Barva šumu	7
1.4 Gabor patch	9
1.4.1 Definice	9
1.4.2 Použití	10
1.5 Entropie	10
1.6 Psychofyzika	11
1.7 Modely pozorovatele	11
1.7.1 d' mapa	12
1.7.2 Modely chování pozorovatele	12
1.8 Percepční učení	14
2 Cíle práce	15
3 Metody	16
3.1 Účastníci	16
3.2 Nástroje a stimuly	16
3.3 Procedura	18
3.4 Limitace	20
4 Měření	22
4.1 Výsledky	22
5 Diskuse	24
Závěr	25
Seznam použité literatury	26
Seznam obrázků	29
Rejstřík	30
Přílohy	31
První příloha – Detailnější výsledky měření	31
Druhá příloha – Dokumentace aplikace	32

Úvod

Prostředí, v němž se lidé každý den pohybují, je komplexní. Abychom se v něm zorientovali, potřebujeme často v našem okolí zrakem najít nějaký objekt. Tato činnost se jmenej zrakové vyhledávání. O zrakovém vyhledávání existují mnohé výzkumy. Jeden z aspektů vizuálního a obecně smyslového vnímání, který se zkoumá, je vztah mezi konkrétními stimuly a jím odpovídajícími vjemy. Tuto závislost zkoumá psychologická disciplína jménem psychofyzika.

Reálný svět je ale obtížně popsatelný. Proto se k experimentům často používají zjednodušené modely, které se v některých aspektech blíží reálným situacím, ale jsou snadno formalizovatelné. Jedna z výhod formálního modelu je možnost zkonstruovat model ideálního pozorovatele. Ten je důležitý pro srovnání úspěšnosti, které dosahuje lidský pozorovatel.

Jedním z úkolů, který se řeší, je vyhledávání konkrétního cíle v poli s vizuálním šumem. Najemnik a Geisler (2005) v nedávné době představili model Ideálního Bayesovského pozorovatele, který modeluje chování lidského pozorovatele s překvapivou přesností. Algoritmus, pomocí kterého Ideální Bayesovský pozorovatel vybíral lokace, které zkoumat, byl ale kubický v počtu možných lokací cíle. Ve svém pozdějším článku (Najemnik a Geisler, 2009) představili pozorovatele ELM, který Ideálního Bayesovského pozorovatele dobře approximuje, a navíc pracuje v kvadratickém čase.

Existence kvadratického algoritmu pro hodnocení kvality otevírá novou možnost: Vzhledem k typickému počtu možných lokací cíle v dané úloze probíhá výpočet jednoho kroku tak rychle, že je z pohledu uživatele okamžitý. Můžeme tedy v reálném čase hodnotit, jak dobré či špatně vůči ELM modelu vybírá lidský pozorovatel lokace, kam se podívá.

Zajímavá otázka je, jakým způsobem se lidský pozorovatel dokáže nejlépe naučit řešit zmíněnou úlohu zrakového vyhledávání (to znamená dosáhnout co nejnižší střední hodnoty počtu pohledů potřebných k nalezení cíle).

Pokud se člověk například bude chtít naučit běhat, nabízí se dva možné přístupy. Budě může pouze sám zkoušet běhat, nebo si může nechat radit od trenéra. V prvním případě má po běhu pouze jedinou zpětnou vazbu: ví, jak se cítí, za jaký čas danou vzdálenost uběhl a podobně. Neví však, jakou měrou za to může jeho běh a jakou náhoda. Trenér naproti tomu může podat například zpětnou vazbu typu „Styl běhu byl nevhodný, a přesto, že se tentokrát nic nestalo, opakováním tohoto postupu může dojít ke zranění“.

Analogicky k tomuto příkladu máme i v naší úloze dva možné přístupy. Učení může probíhat pouze opakováním řešením této úlohy. V takovém případě bude jedinou zpětnou vazbou pozorovatele informace o tom, jak rychle úlohu vyřešil. Druhou možností je, že dostane při každém pohledu informaci, jak dobré či špatně si vybral lokaci podle hodnotící funkce ELM pozorovatele.

V této práci vytvoříme aplikaci, která úlohu zrakového vyhledávání gamifikuje. S její pomocí provedeme experiment, jehož cílem bude zjistit, zda má dodatečná informace z modelu ELM pozorovatele pozitivní vliv na to, jak rychle se účastník po daném počtu pokusů naučí cíl hledat.

Struktura práce

V první kapitole jsou představeny pojmy a teorie, které tato práce využívá, a starší výsledky, z kterých vychází. V druhé kapitole jsou představeny cíle práce. V následující kapitole je popsána metodika experimentu. Poté jsou prezentovány vizualizace naměřených dat, nad nimiž je následně provedena diskuse.

K práci jsou též přiloženy dvě přílohy. V první z nich jsou další grafy z naměřených dat, v druhé se nachází dokumentace aplikace, prostřednictvím které byl prováděn experiment a jejíž vznik byl součástí této práce.

1. Teoretický úvod

V této kapitole přiblížíme některé pojmy a teorie, které jsou v práci využity.

1.1 Zrakové vyhledávání

V úloze zrakového vyhledávání je úkolem najít ve scéně stimul s určitými vlastnostmi. Tyto scény mohou být jednoduché, jako například několik černých úseček na bílém pozadí, kdy je úkolem najít úsečku, která má určitý směr. I takto jednoduché úlohy ale mají mnoho společného se zrakovým vyhledáváním tak, jak ho provádíme v běžném životě (Wolfe a kol., 2011). Mohou ale také být velmi komplikované a těžko popsatelné, jako například hledání muže v pruhovaném oblečení, jaké známe z populární úlohy „Kde je Valda?“.

Předchozí výzkum ukázal, že lidští pozorovatelé v této úloze využívají krátkých pohledů a rychlých pohybů očí (?). Jednomu takovému krátkému pohledu budeme říkat *fixace*. Rychlému očnímu pohybu se říká *sakáda* (Králíček, 2002).

Při vyhledávání nás ve skutečnosti zajímá, ke kterým místům pozorovatel obrací svou pozornost. To však výrazně souvisí s tím, kam se kouká (Posner, 1980).

1.2 Úvod do teorie detekce signálu

Jeden z dílčích problémů úlohy zrakového vyhledávání je rozlišení hledaného objektu a pozadí. K popisu řešení tohoto problému se nám bude hodit terminologie teorie detekce signálu, která se právě touto problematikou zaobírá.

V řeči této teorie budeme hledanému objektu říkat *signál*, rušivým elementům ve scéně se říká *šum*. Problém rozlišení hledaného objektu a pozadí se tedy v řeči teorie detekce signálu formulují jako dotaz, zda se v dané lokaci signál nachází, či nenachází. Tomuto problému se říká Yes-No problém (Macmillan a Creelman, 2004), nebo pouze Y/N.

V kontextu Yes-No problému mohou po odpovědi pozorovatele nastat čtyři možné situace (Green a Swets, 1988):

	Signál je přítomen	Signál není přítomen
Pozorovatel odpoví kladně	<i>Hit</i>	<i>False alarm</i>
Pozorovatel odpoví záporně	<i>Miss</i>	<i>Correct rejection</i>

V této úloze se pozorovatel chová tak, že si zvolí kritérium (někdy též práh odpovědi), které udává, jaká musí být pravděpodobnost, že signál je přítomen, aby pozorovatel odpověděl kladně (Green a Swets, 1988, kapitola 1.7).

Poté pozorovatel provede pozorování, z nějž získá informaci o . Situaci „Pozorovatel získal informaci o “ označíme jako O . Následně spočítá hodnotu rozhodovací proměnné, tedy spočítá, jaká je pravděpodobnost, že je signál přítomen. Tvrzení „Signál je přítomen“ označíme jako S .

Tato pravděpodobnost se z Bayesovy věty spočítá jako

$$P[S|O] = \frac{P[O|S] * P[S]}{P[O]}$$

Potom porovná tuto rozhodovací proměnnou s kritériem a podle výsledku tohoto porovnání odpoví.

Odsud je zřejmé, že zvyšováním kritéria zvyšujeme pravděpodobnost, že nastane miss, ale snižujeme pravděpodobnost false alarmu.

V souvislosti s Y/N úlohou je ale ještě nutné zavést následující pojmy:

Definice 1. Mějme pozorovatele p v úloze Y/N. Potom jeho hit rate H_p definujeme jako

$$P[\text{hit} | \text{Signál je přítomen}]$$

a jeho False alarm rate F_p jako

$$P[\text{False alarm} | \text{Signál není přítomen}].$$

1.2.1 Senzitivita

Jedním s dílčích problémů je též určení, jakou rozlišovací schopnost má daný pozorovatel na daný signál. Zde ale narazíme na problém – není triviální najít parametr, který by rozlišovací schopnost pozorovatele dobře popisoval.

Intuitivně bychom chtěli, aby pozorovatel, který má vysokou senzitivitu, odpovídal ve většině případů správně. Není ale jasné, co to znamená. V předchozí podkapitole jsme ukázali, že pozorovatel často může zvýšit pravděpodobnost hitu za cenu zvýšení pravděpodobnosti false alarmu či naopak. My bychom od senzitivity však chtěli, aby nezávisela na nastavení kritéria.

Proto představíme příklad z praxe, na němž budeme poté ilustrovat výhody a nevýhody různých definic senzitivity.

Nechť je pozorovatelem člověk, který není příliš pozorný a chystá se přejít silnici. Jeho (podvědomým) pozorováním je informace, která dorazí do jeho mozku z periferií jeho sítnic, a signálem, jehož přítomnost zkoumá, je, zda přijíždí auto. V takovém případě se v mozku spustí algoritmus, který převede pozorování na číslo X . Poznamenejme, že v mozku je hodnota tohoto čísla dobře kvantifikovaná, i když je mimo záběr této práce popisovat, co přesně z fyzikálního hlediska tato hodnota znamená. Pro porozumění stačí představa, že se jedná o frekvenci určitého druhu impulsu, které si mezi sebou neurony posílají. Mozek ví, jakou distribuci má náhodná veličina X v případě, že se auto blíží, a jakou má v případě, že nikoliv. Podle konkrétní hodnoty X kterou naměřil, se rozhodne, jestli auto přítomno není, a je tedy bezpečné silnici přejít, nebo zda je pravděpodobnost přijízdějícího auta dost vysoká na to, aby mělo smysl zvednout hlavu a podívat se příslušným směrem znova s cílem získat lepší pozorování.

První intuitivní nápad by byl definovat senzitivitu s jako

$$s = \text{Hit rate}.$$

Z předchozího textu by ale mělo být vidět, že taková definice citlivosti není vhodná – například pozorovatel, který vždy odpoví „ano“ by měl v tomto případě optimální citlivost. V našem případě bychom tedy za každé situace usoudili, že auto přijíždí a silnici bychom nikdy nepřešli, což jistě nechceme.

Na druhý pokus zkusíme senzitivitu nadefinovat jako poměr hitů a false alarmů, tedy jako

$$s = \frac{\text{Hit rate}}{\text{False alarm rate}}.$$

Optimalizace s ohledem k takovému parametru by ale též vedla k sestnému nastavení kritéria.¹

Další možností by bylo charakterizovat pozorovatele jeho poměrem správných a špatných odpovědí. Zapíšeme-li takový parametr ale vzorcem, zjistíme, že by vypadal jako

$$s = \text{Hit rate} * P[S] + (1 - \text{False alarm rate}) * P[\bar{S}] .$$

Taková definice senzitivity též není vhodná, protože potom by citlivost pozorovatele nezávisela jen na jeho vlastnostech, ale i na pravděpodobnosti, že bude signál přítomen.

Praxe ukazuje, že v těchto situacích je rozdelení veličiny X často dobře aproximováno normálním rozdelením, a to jak v případě, že signál přítomen je, tak v případě, že přítomen není. Obě tato normální rozdelení navíc mají v tomto typickém případě stejný rozptyl a liší se jen svou střední hodnotou (Swets, 1996). Potom je přirozenou otázkou ptát se, jaký je rozdíl mezi jejich středními hodnotami (či lépe rozdíl mezi jejich středními hodnotami vydělený směrodatnou odchylkou).

Právě tuto hodnotu označíme jako hodnotu d' . Protože ale ne vždy je rozdelení normální, musíme nadefinovat hodnotu d' způsobem, která nebude závislá na rozdelení veličiny X .

Definice 2. Nechť má pozorovatel p hit rate H_p a false alarm rate F_p , obě ostře mezi jedničkou a nulou. Potom budeme jeho senzitivitu značit d' a spočítáme ji jako

$$d' = \Phi^{-1}(H_p) - \Phi^{-1}(F_p),$$

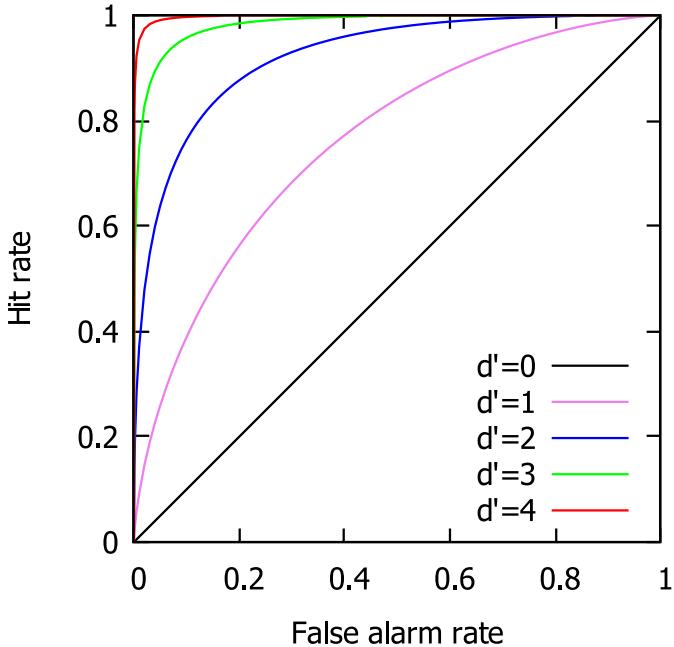
kde Φ^{-1} je inverzní distribuční funkce normovaného normálního rozdelení.

Nahlédneme, že senzitivita má následující vlastnosti:

- Pokud $d' = 0$, jsou indikátorové náhodné veličiny „Pozorovatel odpověděl kladně“ a „Signál je přítomen“ nezávislé.
- Pokud $d' > 0$, mají tyto dvě veličiny kladnou korelacii.
- Pokud skutečně má náhodná veličina X jak v případě, že signál přítomen je, tak v případě, že signál přítomen není, normální rozdelení liší se pouze střední hodnotou, pak hodnota d' nezáleží na nastavení kritéria (samozřejmě krom případů, kdy kritérium nastavíme na plus nebo mínus nekonečno, pak hodnota d' není dobře definovaná).

Závislost hit rate a false alarm rate pro konkretního pozorovatele se nazývá *ROC křivka* z anglického Receiver Operating Characteristic (Swets, 1996; Green a Swets, 1988). ROC křivky pro vybrané hodnoty d' jsou zobrazeny na obrázku 1.1.

¹V typickém případě, který je popsán v následujících odstavcích, se pro kritérium jdoucí k nekonečnu tento poměr též blíží k nekonečnu. Dobrý pozorovatel by tedy téměř vždy odpovídal „ne“, optimální pozorovatel by neexistoval. Kdybychom takové kritérium použili v našem případě, vždy bychom usoudili, že je bezpečné silnici přejít.



Obrázek 1.1: Závislost hit rate a false alarm rate pro různé hodnoty d' . Těmto křivkám se říká ROC křivky z anglického Receiver Operating Characteristic (Swets, 1996; Green a Swets, 1988).

1.3 Šum

V teorii detekce signálu šumem nazýváme nechtěnou (a typicky neznámou) modifikaci signálu. Signály a s nimi spojené modifikace můžou být různé podstaty, například zvukové, nebo se může jednat o šum v elektromagnetickém vlnění. Také se může na signálu projevit mnoha různými způsoby. Může se k němu například přičíst. Takový šum nazveme *aditivní*. Dále existuje například šum multiplikativní či fázový (šum, který se projevuje krátkodobým fázovým posunem signálu).

Pro úlohu zrakového vyhledávání je relevantní aditivní vizuální šum. Nabízela by se otázka, proč nepoužít reálnou scénu, nebo naopak jednobarevné pozadí. Praktické scény jsou však těžko popsatelné a jednobarevné scény (obsahující pouze diskrétní rozptýlení) vnášejí do formalizace vyhledávání nepřesnosti – jejich popis je však nad rámec této práce, lze ho najít v článku (Pelli a Farell, 1999).

1.3.1 Barva šumu

I když však odhlédneme od média, v němž se šum šíří či v něm je zachycený, existuje mnoho různých šumů. Jeden z parametrů, který lze měřit, se nazývá *výkonová spektrální hustota*. Její jednotkou jsou wattys na Hertz a udává, jaký výkon má šum na dané frekvenci. Máme-li hodnoty šumu v dostatečně mnoha bodech, jeho spektrální hustotu spočítáme tak, že na šum aplikujeme Fourierovu transformaci.

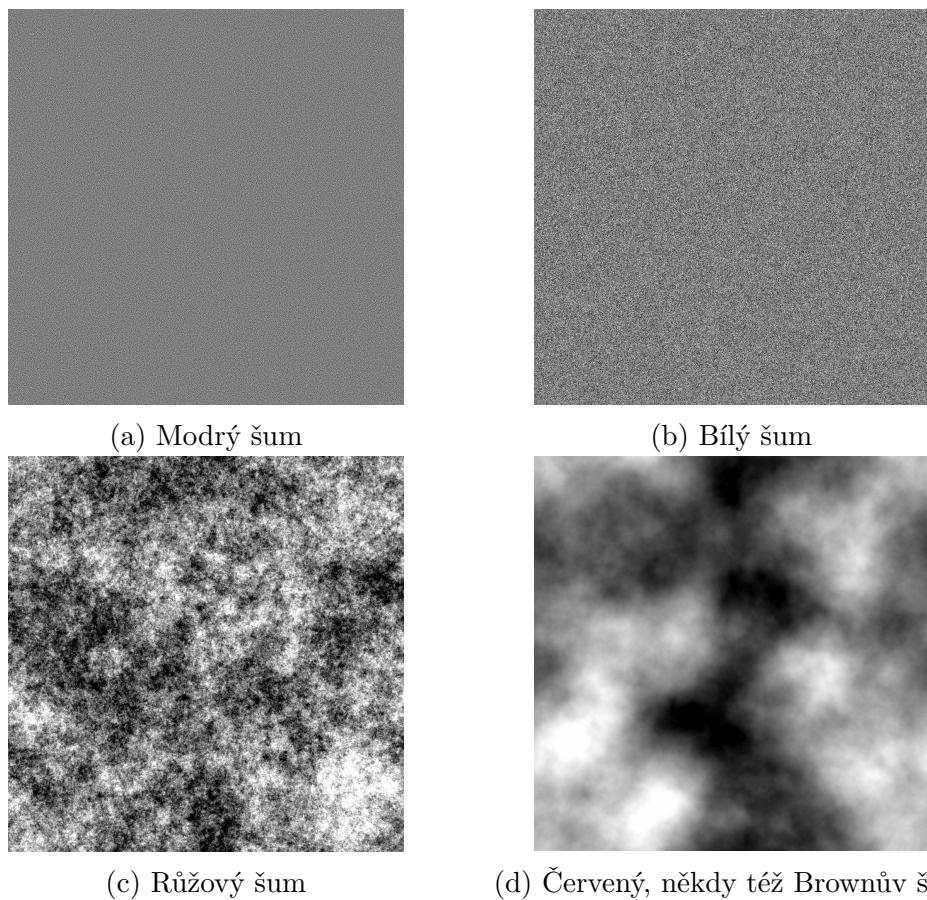
V závislosti na distribuci výkonu napříč různými frekvencemi přiřadíme šumu jméno. Tato jména jsou (i pro jiný než vizuální šum) odvozena od analogie s viditelným světlem – šum se pojmenuje podle barvy, jakou by mělo světlo se stejným rozdělením výkonu napříč viditelným spektrem, jaké má šum rozdělení výkonu

napříč svým spektrem. Takto známe například bílý, růžový, červený (někdy též označovaný jako Brownův či nesprávně hnědý²) či modrý šum (viz obrázek 1.2).

Spektrální hustota p všech zmíněných šumů na dané frekvenci f lze vyjádřit jako $p = 1/f^\beta$, kde hodnota β je -1 pro modrý šum, 0 pro bílý, 1 pro růžový a 2 pro hnědý. Proto se růžový šum někdy též označuje jako $1/f$ šum. Pro ostatní barvy šumu není podobné označení běžné. O šumu a jeho barvách je pojednáno více v knize Noise (Kosko, 2006).

Pravděpodobnostní rozdělení jednotlivých složek Fourierovy transformace ale není definicí barvy šumu dáno. Pokud je rozdělení normální (Fourierova obrazu nebo samotného šumu), řekneme, že se jedná o Gaussovský šum.

V této práci se budeme zabývat vizuálním šumem, tedy šumem, kde místo obvykle používané časové souřadnice použijeme dvě souřadnice prostorové a měřenou hodnotou bude jas.



Obrázek 1.2: Ukázky různých šumů. Tyto šumy byly vygenerovány pomocí algoritmu použitého přímo v aplikaci, která je součástí této práce. Algoritmus je blíže popsán ve druhé příloze.

²V anglické literatuře se používá termín *Brownian noise*, setkáme se ale i se zavádějícím termínem *Brown noise*. Jméno je odvozeno od souvislostí s Brownovým pohybem částic

1.4 Gabor patch

V úloze zrakového vyhledávání potřebujeme také nějaký cíl. V reálném životě vnímáme mnoho různých podnětů. Pro takovou situaci by však bylo příliš obtížné najít model. Proto se často používají jednodušší, lépe popsatelné cíle. Jedním z takových stimulů je *Gabor patch* vycházející z Gabor filteru.

Gabor filter (v českých textech někdy označovaný jako Gaborova vlnka) je lineární filtr používaný ve zpracování obrazu, chceme-li detektovat signál mající danou frekvenci a směr, který se vyskytuje kolem daného bodu.

1.4.1 Definice

Hodnotu filtru v daném bodě spočítáme jako součin dvou funkcí. První z nich nazveme *jádro* filtru. Jako jádro se vždy používá sinus či cosinus (někdy uváděné v podobě komplexní exponenciály, pokud potřebujeme i reálnou, i imaginární složku). Jeho parametry určují, jaké vlastnosti má mít signál, který chceme detektovat. Druhé funkci říkáme *obálka*, a určuje, na jakém okolí daného bodu signál zkoumáme.

Funkci tedy lze vyjádřit jako

$$g(x,y) = \sin\left(2\pi\frac{x'}{\lambda} + \phi\right) * \text{obálka}(x',y'),$$

kde vektor (x',y') je vektor (x,y) otočený o úhel, který svírá osa x se směrem, podél nějž chceme měřit signál (tentou úhel budeme značit Θ), a posunutý do bodu, v němž chceme měřit signál, λ je frekvence signálu, který hledáme, a ϕ je fázový posun (Fogel a Sagi, 1989). Frekvence signálu se nejčastěji udává v cyklech na jednotku vzdálenosti, často na pixel obrazu.

Jako obálka se používá dvojrozměrná Gaussova funkce, raised cosine, nebo prostá lineární funkce vzdálenosti.

Gaussovou funkci vyjádříme jako

$$\text{obálka}(x,y) = \exp\left(\frac{x'^2 + y'^2}{2\rho}\right),$$

kde ρ je směrodatná odchylka Gaussovy křivky. Její výhodou je, že chování Gabor filtru, jehož obálku tvoří Gaussova funkce, je nejpřesněji popsané. Raised cosine vyjádříme jako

$$\text{obálka}(x,y) = \begin{cases} \frac{\cos(\pi\sqrt{x'^2+y'^2}/r)+1}{2} & \text{pro } \sqrt{x'^2+y'^2} \leq r, \\ 0 & \text{jinak,} \end{cases}$$

kde r je poloměr oblasti, v níž chceme signál detektovat. Výhodou raised cosine oproti Gaussově funkci je, že ve vzdálenosti alespoň r od středu filtru jeho hodnota nabývá nuly. Při výpočtech tedy stačí počítat s malou oblastí kolem středu (kdežto při použití Gaussovy funkce je nutné počítat s celým obrazem). Výhodou oproti lineární funkci vzdálenosti je, že raised cosine se pro většinu aplikací chová dostatečně podobně jako Gaussova funkce.

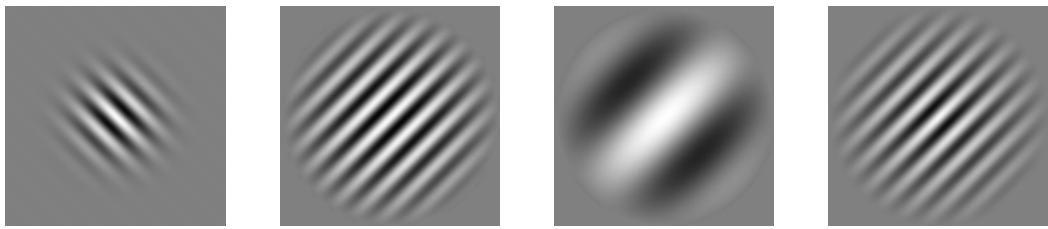
1.4.2 Použití

Chceme-li detekovat signál ve vizuálním šumu, spočítáme hodnotu

$$s = \sum g(x,y) * n[x,y],$$

kde n je šum a sumu bereme přes všechny body (x,y) , v nichž jsme naměřili hodnoty šumu. Je-li hodnota s blízko nuly, signál v daném místě není přítomen, nebo je přítomen s jinými parametry. Vysoké hodnoty značí, že signál pravděpodobně přítomen je, hluboce záporné značí, že signál je přítomen, ovšem s fází posunutou π .

Gabor filter ale můžeme používat i k samotné tvorbě signálu. Chceme-li vytvořit v daném bodě signál, můžeme spočítat Gabor filter, jako bychom chtěli detekovat signál s právě takovými parametry, jaké má mít tvořený signál, a potom ho sečíst se šumem. Takto vytvořenému signálu budeme říkat Gabor patch (několik příkladů viz obrázek 1.3).



Obrázek 1.3: Ukázky několika Gabor patchů. Všechny Gabor patche jsou 100 pixelů široké i vysoké. Levý patch má $\Theta = 1/4\pi$, ostatní mají $\Theta = -1/4\pi$, levý má jako obálku Gaussovou funkci, prostřední dva raised cosine, pravý lineární funkci vzdálenosti, první, druhý a čtvrtý patch mají frekvenci (v cyklech na pixel) 0.1, třetí 0.02.

1.5 Entropie

Další pojem, který budeme v této práci potřebovat, pochází z teorie informace. Entropií (nazývanou též Shannonova entropie) náhodné veličiny se v této teorii rozumí střední hodnota informace, kterou nám hodnota této veličiny přinese. Například mějme náhodné veličiny X a Y . X nechť nabývá hodnoty 0 s pravděpodobností 1/2 a hodnoty 1 s toutéž pravděpodobností. Y nechť nabývá hodnoty 1 s pravděpodobností 1 a hodnoty 0 s pravděpodobností 0. Je vidět, že pokud se dozvíme, jaké hodnoty nabývá veličina X , získáme více informace, než když zjistíme, jaké hodnoty nabývá Y .

Hodnotu informace, kterou nám přineslo zjištění, že náhodná veličina A nabývá hodnoty a , spočítáme jako

$$I(A = a) = -\log_b (\text{P}[A = a]) = \log_b \left(\frac{1}{\text{P}[A = a]} \right).$$

Jako základ logaritmu b se běžně používá 2 (což budeme dodržovat i v této práci), e nebo 10. Všimneme si, že toto vyjádření množství informace je konzistentní s intuitivní představou, že pokud A nabývá nějaké nepravděpodobné hodnoty, je

informace o této hodnotě cennější, než kdyby nabývalo nějaké pravděpodobné. Odsud tedy entropii $H(A)$ lze vyjádřit jako

$$H(A) = \mathbb{E}[I(A)] = - \sum_{a \in \Omega} P[A = a] \log_2 P[A = a],$$

kde $\Omega = \{a | P[A = a] > 0\}$. Všimneme si, že podle tohoto vzorce je entropie veličiny X , tak jak byla nadefinována v předchozím odstavci, rovna jedné, kdežto entropie veličiny Y je rovna nule.

Podobně jako u pravděpodobnosti můžeme poměrně přímočaře nadefinovat i podmíněnou entropii $H(A|B)$

Shannonova entropie byla v této podobě zavedena Claudem Shannonem v jeho článku (1948).

1.6 Psychofyzika

Psychofyzika je disciplína psychologie, která zkoumá vztah mezi konkrétními stimuly a jim odpovídajícími vjemy. Lze ji aplikovat na libovolný smyslový vjem, ať už obraz, zvuk, vůni, chuť či hmatový vjem (Kingdom a Prins, 2009). Funkcí, které dávají do souvislosti parametry stimulů a chování subjektu v experimentu, se říká *psychometrické funkce*.

Při detekci signálu rozeznáváme dva druhy šumu. Šum vnitřní a šum vnější (Heeger, 1997). Vnější šum je zkreslení dat, které nastane ještě mimo pozorovatele. Vnitřní šum je naopak šum způsobený přímo algoritmem, který používá pozorovatel, aby dekódoval, zda signál je přítomen, či není.

Vrátíme se zpět na začátek této kapitoly k příkladu s chodcem přecházejícím silnicí. Vnější šum tam může být způsoben množstvím faktorů. Například může nějaký předmět částečně blokovat výhled. Nebo může být chodcův zrak nedokonalý. Nebo se může směrem, ze kterého by se mohlo blížit auto, nějaký jiný předmět, který vypadá trochu jako auto. Vnitřní šum je naopak čistě uvnitř mozku, kdy mozek vjem, u kterého k tomu nemá žádný důvod vyhodnotí jako přijíždějící auto.

Právě chování tohoto vnitřního šumu zkoumá psychofyzika.

1.7 Modely pozorovatele

V teorii detekce signálu je velmi důležitým konceptem ideální detektor. Mohlo by se nám zdát, že by nás ideální pozorovatel nemusel zajímat – pracujeme přece s pozorovatelem lidským. Pro lidského pozorovatele ale potřebujeme srovnání. V případě, kdy je algoritmus ideálního pozorovatele v určitém smyslu přirozený, například nedělá žádné kontraintuitivní kroky, můžeme také tvrzení „Lidský pozorovatel se chová ideálně“ použít jako úvodní hypotézu, kterou poté už pouze upravujeme či zpřesňujeme (Green a Swets, 1988).

V této práci se budeme zabývat úlohou, kdy pozorovatel hledá Gabor patch v kruhovém poli, v němž se nachází růžový šum.

Abychom mohli hodnotit fixace lidského pozorovatele, potřebujeme mít nějaký model, který nám bude říkat, jak se chová optimální pozorovatel v této úloze. Nejprve ale stručnou odbočku:

1.7.1 d' mapa

Ještě předtím, než začneme zjišťovat, podle jakého modelu se chová lidský pozorovatel, je nutné zjistit, kolik informace člověk jednou fixací získá. Je například zjevné, že kdyby jedním pohledem bez ohledu na to, kam se dívá, získal stejné množství informace o všech možných polohách cíle, jsou všechny modely ekvivalentní. Proto je potřeba najít tzv. d' mapu, funkci, která nám pro každé dva body x, y řekne, jakou hodnotu d' má lidský pozorovatel, pokud hodnotí pozici y a dívá se na pozici x . U pozadí s uniformními nebo téměř uniformními lokálními kontrasty lze tuto funkci zjednodušit, stačí, když budeme pro každý bod x vědět, jaká je hodnota d' pro rozhodování, zda je cíl v bodě x , pokud zafixujeme střed.

Jedná se tedy o psychometrickou funkci, kde je parametrem poloha stimulu, a chování subjektu je vyjádřeno jeho hodnotou d' .

Předchozí výzkum ukázal, že u běžného člověka je d' mapa poměrně nepravidelná (Najemník a Geisler, 2008), ale dá se s přijatelnou přesností approximovat mapou, kde křivky spojující body se stejnou hodnotou d' jsou tvořeny čtyřmi čtvrtelipsami (jednou v každém kvadrantu) se středem v počátku, s excentricitami a velikostí poloos danou hodnotou d' a individuálními vlastnostmi pozorovatele a cíle (Bradley a kol., 2014).

Ke kompletnímu vyjádření approximace funkce d' potřebujeme celkem 6 hodnot, které závisí na konkrétním pozorovateli a cíli. Jedná se o hodnotu $d'_0 = d'(0,0)$, hodnot e_L, e_R, e_U a e_D , které určují vzdálenost ve čtyřech základních směrech takovou, že v nich je hodnota d' poloviční oproti počátku, a hodnotu β popisující sklon této funkce. Funkci d' pak vyjádříme jako

$$d'(x,y) = \frac{d'_0}{1 + \left(\frac{x^2}{e_H^2} + \frac{y^2}{e_V^2} \right)^\beta},$$

kde e_H je rovno e_L pro záporná x a e_R pro kladná, a e_V je rovno e_D pro záporná y a e_N pro kladná.

1.7.2 Modely chování pozorovatele

Všechny modely, které zde budeme zkoumat, vypadají tak, že mají tzv. *Mapu aposteriorních pravděpodobností*. V této mapě je pro každou lokaci, kde by signál (též cíl) mohl být, uvedeno, jaká je pravděpodobnost, že se na ní cíl nachází vzhledem k informaci, kterou již o dané lokaci pozorovatel získal.

Pozorovatel postupuje tak, že si vždy vybere lokaci, kterou zafixuje. Poté provede fixaci a získá o každé lokaci určitou informaci. Pak spočítá novou mapu posteriorních pravděpodobností. Pokud na některé lokaci pravděpodobnost přesáhne kritérium, pozorovatel ukončí hledání a ohlásí nalezení cíle na této lokaci. Jinak pokračuje ve vyhledávání výběrem další lokace k fixaci. Jednotlivé modely se dále liší pouze algoritmem, který používají k výběru lokace k fixaci.

V rámci předchozího výzkumu bylo otestováno mnoho modelů, jako například pozorovatel, který volí fixace náhodně, nebo pozorovatel, který volí fixace tak, aby nikdy nezafixoval tutéž lokaci vícekrát. Všechny tyto modely se ale ukázaly jako nevhodné vzhledem k tomu, že v praxi dosahují mnohem horších výsledků (měřeno pomocí střední hodnoty počtu fixací před nalezením cíle) než lidský pozorovatel. (Najemník a Geisler, 2005).

MAP pozorovatel

Nejjednodušší model, který dosahuje podobných výsledků jako lidský pozorovatel, je tzv. MAP³ pozorovatel, který vždy zafixuje lokaci, která má v jeho mapě aposteriorních pravděpodobností nejvyšší hodnotu. Tento pozorovatel již dosahuje podobných výsledků jako lidský, ale jeho strategie fixací neodpovídá strategii, jakou volí lidský pozorovatel. Lidský pozorovatel umístí svoji první fixaci do středu scény. Ostatní fixace jsou pak rozmístěny v okolí kružnice se středem ve středu scény a poloměrem rovným přibližně 2/3 poloměru scény, s preferencí pro horní a spodní okraj. MAP pozorovatel oproti tomu vybírá každou lokaci se zhruba stejnou pravděpodobností.

Ideální Bayesovský pozorovatel

Ještě o trochu lepších a hlavě statisticky lidskému pozorovateli bližších výsledků dosahuje model Ideálního Bayesovského pozorovatele.

Ideální Bayesovský pozorovatel (dále IBO) je pozorovatel, který $T + 1$ lokaci vybírá tak, aby maximalizoval pravděpodobnost, že v následujícím kroku odhalí cíl. Vybere tedy lokaci

$$k_{opt}(T + 1) = \arg \max_{k(T+1)} \left(\sum_{i \in L} p_T(i) P[p_{T+1}(i) \geq c | i, k(T + 1)] \right), \quad (1.1)$$

kde p_N je mapa aposteriorních pravděpodobností po N -té fixaci, L je množina všech potenciálních lokací cíle a c je kritérium, které musí hodnota v mapě aposteriorních pravděpodobností překročit, aby bylo ukončeno hledání a nahlášeno nalezení cíle. Výraz $P[p_{T+1}(i) \geq c | i, k(T + 1)]$ pak tedy znamená „pravděpodobnost, že po $T + 1$ fixaci bude ukončeno hledání a nahlášen signál v lokaci i , za podmínky, že v ní signál opravdu je přítomen a byl zafixován bod $k(T + 1)$ “.

IBO však ale též není příliš dobrým kandidátem na model, podle nějž se lidé chovají. Ač má jeho vyhledávání podobné statistické vlastnosti jako vyhledávání lidského pozorovatele, vyžaduje perfektní pamět a ideální integraci informace mezi fixacemi. Ani jednu z těchto dovedností ale lidský pozorovatel nemá. Dále si můžeme všimnout, že přinejmenším přímočaré vyhodnocení výrazu (1.1) je kubické v počtu potenciálních lokací (výpočet druhého činitele součinu v sumě je lineární, suma sama je přes lineárně mnoho členů a vnější maximum má též lineárně mnoho možných voleb $k(T + 1)$). Lidský pozorovatel při vyhledávání volí další fixaci přibližně třikrát až čtyřikrát za vteřinu, je tedy nepravděpodobné, že by lidský mozek dokázal takový výpočet provést (Najemník a Geisler, 2008).

ELM pozorovatel

ELM⁴ pozorovatel je pozorovatel, který při výběru následující fixace minimizuje střední hodnotu entropie náhodné veličiny určující lokaci cíle po fixaci. Tu spočítáme jako

$$\mathbb{E}[H(T + 1) | k(T + 1)] = -\mathbb{E} \left[\sum_{i \in L} p_{T+1}(i) \log_2 p_{T+1}(i) | k(T + 1) \right]. \quad (1.2)$$

³Z anglického „Maximum Aposteriori Probability“.

⁴Z anglického „Entropy limit minimization“.

Najemník a Geisler (2009) však ukázali, že vyjádření hodnoty entropie podle vzorce (1.2) je sice netriviální, ale lze ho dobré approximovat případem, kdy posleme v limitě $|L|$ do nekonečna. Pak dostáváme výraz

$$\mathbb{E} [\text{H}(T+1)|k(T+1)] = \text{H}(T) - \sum_{i \in L} p_T(i) d'^2(i - k(T+1)),$$

kde v posledním členu bereme lokace jako vektory od počátku k nim. Člen $\text{H}(T)$ navíc nezávisí na $k(T+1)$, takže s ním vůbec nemusíme počítat a k minimalizaci entropie nám stačí maximalizovat hodnotu sumy.

1.8 Percepční učení

TODO

2. Cíle práce

Cílem této práce je zjistit, zda se lidé dokáží naučit hledat konkrétní Gabor patch v kruhovém růžovém šumu lépe, pokud jim při tréninku budeme po každé fixaci dávat akusticky informaci o tom, jak výhodné bylo zvolit k fixaci právě bod, který zvolili. Jejich schopnost najít cíl budeme hodnotit průměrným počtem fixací potřebných k nalezení cíle. Tohoto cíle dosáhneme následujícími kroky:

1. Vytvoříme aplikaci pro operační systém iOS, v níž bude uživatel hledat Gabor patch v kruhovém poli a volitelně bude dostávat zpětnou vazbu, jak byla popsána v předchozím odstavci.
2. Navrhneme experiment, během nějž budeme pomocí zmíněné aplikace nejprve měřit schopnost účastníků hledat Gabor patch, poté se budou účastníci učit hledat Gabor patch a nakonec opět tuto schopnost změříme.
3. Najdeme několik účastníků, které náhodně rozdělíme do dvou skupin, kontrolní a experimentální. Všechny účastníky necháme projít experimentem navrženým v předchozím bodě. Účastníci v kontrolní skupině se budou učit hledat Gabor patch pouze tím, že ho budou opakovaně hledat. Účastníci v experimentální skupině budou navíc ve fázi učení dostávat zpětnou vazbu.
4. Výsledky experimentu zpracujeme běžnými statistickými metodami, aby chom zjistili, zda se účastníci v experimentální skupině zlepšili během učení výrazněji než účastníci v kontrolní skupině, a pokud ano, tak zda je rozdíl mezi skupinami statisticky významný.

3. Metody

V této kapitole popíšeme metody, kterými byl prováděn experiment. Technické detaily, jako například algoritmus, kterým byl generován šum, nebo hodnoceny lokace, jsou ale uvedeny až v dokumentaci aplikace, která je druhou přílohou této práce.

3.1 Účastníci

Experiment byl proveden celkem s 10 účastníky, pěti v každé skupině. Z ruk všech z nich byl normální nebo upravený na normální (např. brýlemi). Byli vybráni z okolí autora této práce, nejednalo se tedy o náhodný výběr z populace. Do skupin byli rozřazeni náhodně. Ve skupině se zpětnou vazbou byl věkový průměr 26.8 let, směrodatná odchylka věku byla 10.1, dva z účastníků byli muži a tři ženy. Ve skupině bez zpětné vazby byl průměrný věk 26 let se směrodatnou odchylkou 11.1 let, jeden muž a čtyři ženy. Detailnější statistiky jsou uvedeny v první příloze. Za účast na experimentu nebyli účastníci nijak odměněni a účastnili se ho dobrovolně. Na začátku experimentu žádný účastník nevěděl nic o vlastnostech modelů IBO a ELM.

3.2 Nástroje a stimuly

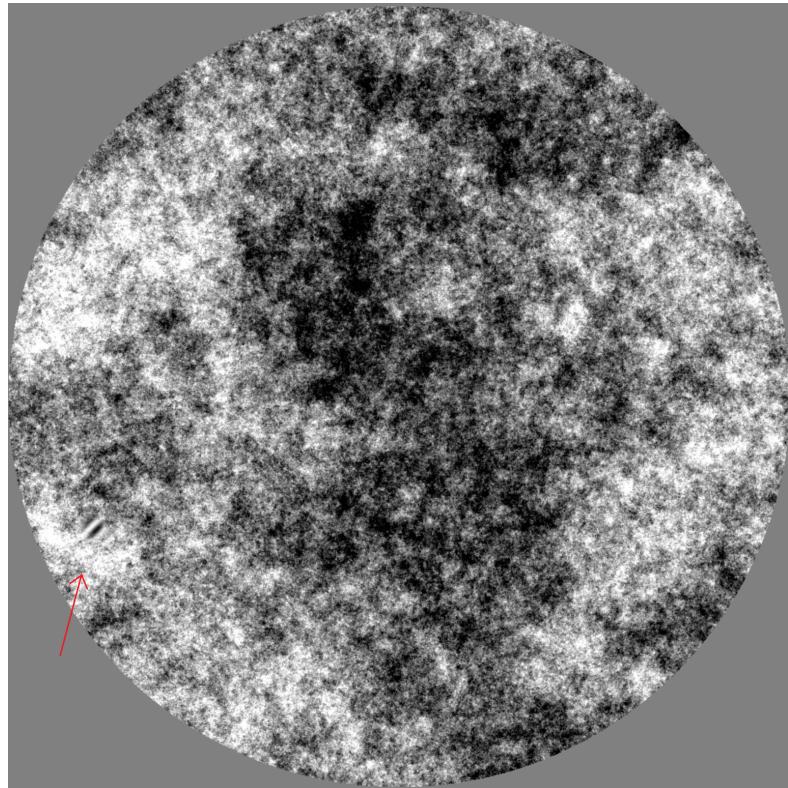
K experimentu byla použita aplikace, která je součástí této práce. Jako zobrazovací zařízení byl použit iPad air s displejem o rozlišení 2048×1536 a o úhlopříčce 9.7 palců, což odpovídá rozměru displeje asi 19.7×14.8 centimetrů. Hustota pixelů je 264 pixelů na palec.

V experimentu použitý růžový šum byl kruhový a jeho průměr byl 1024 pixelů¹. Tento průměr byl zvolen proto, že je to nejbližší mocnina dvojkdy ke kratšímu rozměru displeje iPadu v pixelech. Mocnina dvojkdy byla zvolena kvůli použití rychlé Fourierovy transformace při generování růžového šumu. RMS kontrast² šumu byl po vygenerování roven 0.25, ale při zobrazování byly hodnoty lišící se od střední hodnoty o více než dvě standardní odchylky přiblíženy ke střední hodnotě právě na tuto vzdálenost, čímž byl RMS kontrast mírně snížen (pro příklad šumu takového, jaký byl použit viz obrázek 3.1).

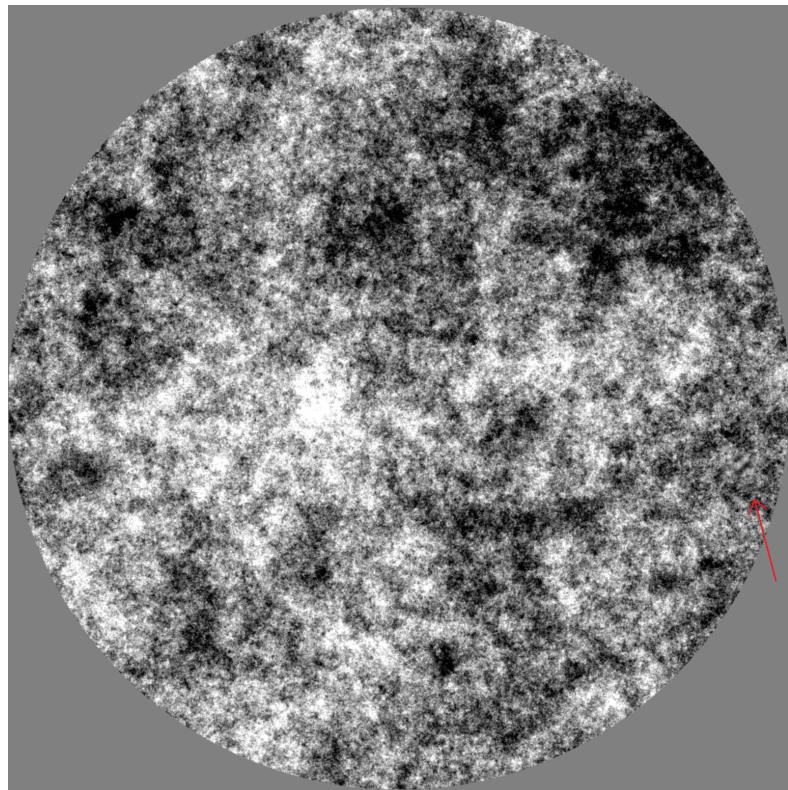
Jako stimulus byl použit Gabor patch. Jedním z problémů aditivního Gabor patche ale je fakt, že jas pixelu je v praxi omezený. Kdybychom tedy přičítali Gabor patch k šumu v místě, které má samo o sobě vysoký jas, museli bychom jeho nejvyšší bod snížit tak, aby součet se šumem nepřesáhl maximální hodnotu jasu pixelu. Obdobný problém bychom měli s oblastí s nízkým jasem. Tento problém byl vyřešen tak, že Gabor k šumu nebyl ale přičten k šumu, ale vložen do něj, jako kdybychom kreslili Gabor patch přes šum a obálka zastupovala alfa kanál. To znamená, že jas pixelu v bodě x byl spočítán jako $S[x] * (1 - o[x]) + G[x] * o[x]$,

¹Odsud již všude, kde není zřejmý opak, se pixelem myslí pixel obrázku, nikoliv pixel displeje.

²RMS kontrast je pro černobílý obraz vlastně jiný název pro standardní odchylku jasu pixelu, kdy jas měříme tak, aby černý pixel dostal hodnotu nula a bílý hodnotu jedna (Peli, 1990).

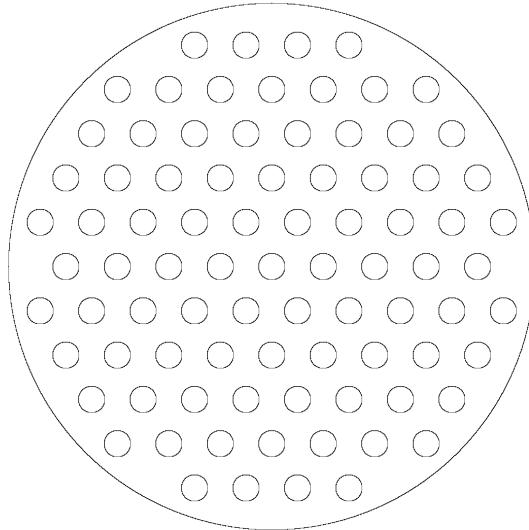


(a) Šum s dobře viditelným Gabor patchem u levého dolního okraje (kontrast 0.99).



(b) Šum se špatně viditelným Gabor patchem u pravého okraje (kontrast 0.61).

Obrázek 3.1: Příklady šumů s cíli.



Obrázek 3.2: Možné lokace Gabor patche

kde $S[x]$ je hodnota šumu v bodě x , a $G[x]$ a $o[x]$ jsou hodnoty Gaboru a jeho obálky. Zvolené parametry Gabor patche byly:

- Obálka: Raised cosine
- Průměr: 50 pixelů
- Frekvence: 1/16 cyklu na pixel
- Fázový posun: 0
- Úhel Θ : 135°

Možných lokací Gabor patche bylo celkem 85, a byly rozmístěny po scéně v trojúhelníkové mřížce tak, aby jedna možná lokace byla ve středu. Vzdálenost dvou sousedních možných lokací byla 100 pixelů (viz obrázek 3.2).

Kontrast cíle byl daný maximem obálky, tedy při snižování kontrastu byl cíl čím dál tím průhlednější. Hodnoty kontrastu se mohly pohybovat mezi nulou a jedničkou.

3.3 Procedura

Každý účastník prošel sadou 3 testů. V prvním testu mu bylo postupně prezentováno 40 úkolů, kde v každém z nich měl najít Gabor patch v růžovém šumu. V druhém testu bylo prezentováno 120 obdobných úkolů a ve třetím opět 40.

Ve druhém testu dostávaly účastníci, kteří byly ve skupině se zpětnou vazbou, po každé fixaci zvukovou odpověď, která značila, kolik informace mohli od této fixace očekávat (tedy jestli bylo z pohledu ELM moudré udělat právě tuto fixaci). Tato odpověď byla ve formě tónu, jehož frekvence f byla dána vzorcem

$$f = 440 \text{ Hz} * 2^{2 - 2 * \frac{c - \delta}{\Delta - \delta}},$$

kde c je očekávané snížení entropie při fixaci, kterou si subjekt vybral, Δ je maximální dosažitelné očekávané snížení entropie a δ minimální v případě, že by byla zafixována některá z možných lokací cíle.³

Tento vzorec byl sestaven tak, aby účastník dostal tón a1 v případě, že zvolil lokaci, která vedla ke stejnemu očekávanému snížení entropie, jako lokace, kterou vybral model ELM jako nejlepší, tón a3 v případě, že naopak zvolil nějakou z nejhorších možných lokací.

V každém úkolu byl šum překryt černou barvou. Subjekt se měl vždy dotknout displeje v místě, které se rozhodl zafixovat. Na tomto místě byl poté šum odkryt na 300 ms. Výpočet tvaru a míry odkryté oblasti bylo provedeno vynásobením s d' mapou posunutou do bodu fixace, s parametrem d'_0 nastaveným na 1 a ostatními parametry naměřenými na pozorovateli FD ($e_R = 223$, $e_L = 223$, $e_U = 161$, $e_D = 164$, $\beta = 2.46$, všechny veličiny, u nichž má smysl uvádět jednotku, jsou v pixelech).

Tuto d' mapu by bylo lepší měřit každému účastníkovi zvlášť. Takové měření ale zahrnuje podle metodiky z článku (Bradley a kol., 2014) přinejmenším 2000 trialů a s metodikou předepsanými přestávkami trvá celý den. Aby bylo možné při něm kontrolovat chování účastníka, je navíc potřeba eyetracker. Proto jsme se rozhodli i vzhledem k nemožnosti kontrolovat přesně podmínky k tomuto měření nepřistoupit. K tomuto problému se ale znova vrátíme závěrem této kapitoly v sekci Limitace.

Ve chvíli, kdy si účastník myslí, že objevil cíl, zmáčkl tlačítko. Poté mu byl ukázán celý odkrytý šum, ovšem bez cíle. Potom se měl účastník dotknout šumu na místě, kde si myslí, že se cíl nacházel. Cíl byl považován za nalezený, pokud byla vzdálenost vybraného místa a středu skutečné lokace cíle menší než 60 pixelů displeje. Úkol byl považován za úspěšně splněný, pokud byl cíl nalezen a současně v rámci něj účastník provedl nejvýše šest fixací.

V každém testu byl počáteční kontrast cíle 0.7. Pokaždé, když byl účastník třikrát po sobě úspěšný, byla zvýšena obtížnost snížením kontrastu cíle o 0.01, pokud byl třikrát po sobě neúspěšný, byla obtížnost opět snížena.

Byl tedy využit obecný postup, kterému se říká Up/Down metoda. Tento postup se používá, pokud je závislost pravděpodobnosti daného jevu na nějakém parametru rostoucí (či obecně monotónní) funkce. Spočívá v tom, že se daný parametr snižuje, když jev nastává, a zvyšuje když nenastává (v obou případech o předem danou konstantu, která se během experimentu nemění, nemusí však být v obou směrech stejná). Blíže je popsána v knize Psychophysics: A practical introduction (Kingdom a Prins, 2009). Na rozdíl od implementace této metody, která je popsaná v knize, jsme se rozhodli za jev považovat tři úspěchy za sebou a za jeho absenci tři neúspěchy, abychom zmenšili velký rozptyl, který náš jev jinak má.

Hodnota 0.01 byla vybrána tak, aby se během 120 úkolů, které jsou v prostředním testu, účastník mohl teoreticky dostat až na hodnoty kontrastu kolem 0.3. S kontrastem menším než přibližně 0.45 je ale pravděpodobnost splnění úkolu bez ohledu na strategii podle subjektivního názoru autora práce velmi nízká. V experimentu se tento dojem potvrdil, nejnižší v experimentu dosažený kontrast byl 0.5.

³To znamená, že je potenciálně možné dosáhnout výsledku lepšího než Δ , resp. horšího, než δ . Rozdíl by však neměl být důležitý.

3.4 Limitace

Při návrhu experimentu jsme narazili na několik problémů, které mohou vnést nepřesnosti do měření, ale jejichž řešení je mimo rozsah této práce. Konkrétně se jedná o následující obtíže:

- Vjemy lidského pozorovatele neodpovídají příliš dobře vjemům simulovaného ideálního pozorovatele. Aby si tyto vjemy odpovídaly, alespoň přibližně, museli bychom každému pozorovateli změřit jeho vlastní d' mapu. Měření d' mapy ale i v té nejminimalističtější variantě, která se používá, trvá nejméně jeden pracovní den. Druhým důvodem, proč mohou být vjemy rozdílné i v případě, že by konstanty d' mapy vyšly pozorovateli stejně, jaké byly použity, je, že tato naměřená d' mapa odpovídá situaci, kdy je scéna se šumem umístěna tak daleko od pozorovatele, aby ji viděl pod zorným úhlem 15° . To při velikosti scény v našem případě odpovídá vzdálenosti pozorovatele a zařízení přibližně 65 cm. V našich experimentech nebyla vzdálenost pozorovatele od scény hlídána a určitě byla nižší, než řečených 65 cm (do držení této vzdálenosti by odpovídalo situaci, kdy by účastníci držely iPad před sebou zhruba na délku natažené paže).
- I pokud odhlédneme od nepřesností zmíněných v předchozím bodě a dovolíme si na chvíli (evidentně sestný) předpoklad, že účastníci měly vlastní d' mapu konstantní, narazíme na další problém. Okraj oblasti, která byla odkrytá, byl ztmavován lineárně se snižující se hodnotou d' v použité d' mapě. Závislost d' na kontrastu ale téměř jistě není lineární.
- S tím souvisí ještě jeden problém: Subjekty samozřejmě nemají svou d' mapu konstantní. Tato mapa se tedy nějak skládá s d' mapou, pomocí které bylo určeno odhalování šumu. V práci jsme toto skládání ignorovali (tedy předpokládali jsme, že d' na kontrastu závisí lineárně a d' mapa účastníků je konstantní.) Nabízela by se otázka, proč tedy bylo zatemňování šumu vůbec prováděno. To se dělo z několika důvodů:
 - Zatemňování šumu zavádí potřebu klikat na místa, která chce pozorovat v dalším kroku prozkoumat. Nutí tedy pozorovatele, aby tato rozhodnutí dělal vědomě a nikoli podvědomě, což byl jeden z efektů, které jsme chtěli zkoumat.
 - Celý proces jedné fixace tímto způsobem také trvá mnohem déle (nižší jednotky vteřin místo nižších desetin vteřiny) a poskytuje nám tedy mnoho času na update mapy aposteriorních pravděpodobností a výpočet množství informace, kterou lze získat následující fixací.
 - Takto navržený experiment též umožňuje zjišťovat, které lokace účastník fixuje bez použití eyetrackeru nebo jiných technologií.
- Je možné, že u lidí, kteří nikdy dříve Gabor patch hledat nezkoušeli, se během testů mění jejich senzitivita na Gabor patch (mění se hodnota $d'(0,0)$).
- Vzhledem k tomu, že počet reálných pixelů displeje neodpovídá (a ani nebyl dělitelný) velikostí scény v pixelech, je možné, že byly efekty jako například antialiasing změněny lokální kontrasty scény.

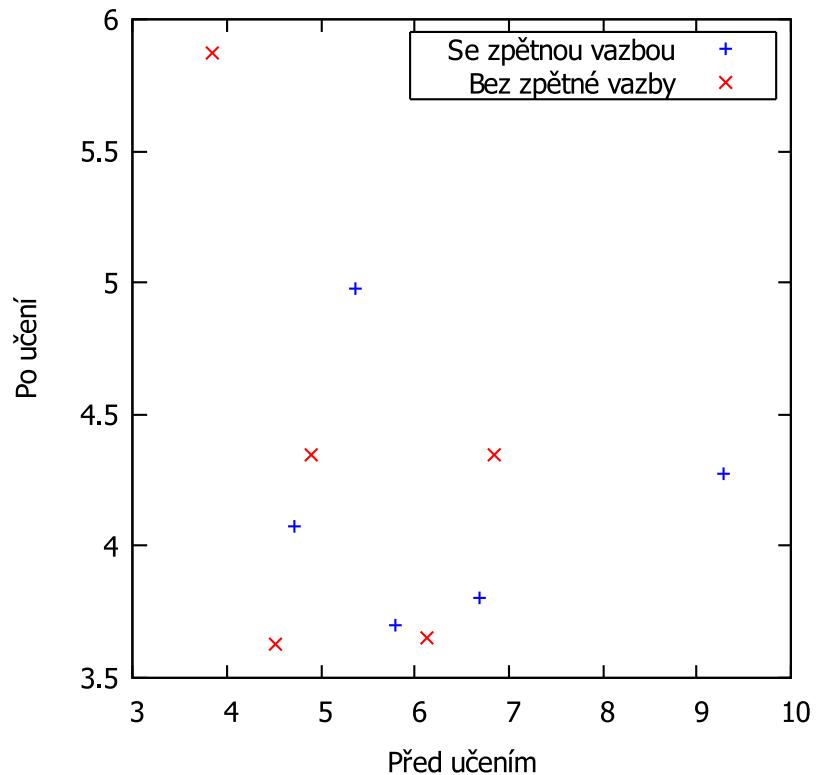
- Ve výzkumu v oblasti psychofyziky se většinou pečlivě kontroluje prostředí (například se zatemňuje místnost, v níž se provádí experiment). To jsme v našem výzkumu nedělali.

Tyto nepřesnosti však nepovažujeme za příliš důležité, protože cílem této práce není přesný experiment, ale pouze proof of concept. Za zásadní chybu naopak nepovažujeme malý počet účastníků – kdybychom chtěli na této práci postavit přesný experiment, nebylo by potřeba zvyšovat počet účastníků, ale pouze počet měření na jednotlivých účastnících, například tak, že bychom vícekrát opakovali první a třetí test. Jinak ale není ve výzkumu nutně přínosné velký vzorek, často je lepší na malém vzorku provést větší počet přesnějších měření (Smith a Little, 2018). Kdybychom se však rozhodli raději pro Large-*N* design, mohli bychom většinu ostatních parametrů experimentu ponechat, ale bylo by potřeba přinejmenším o řád více účastníků.

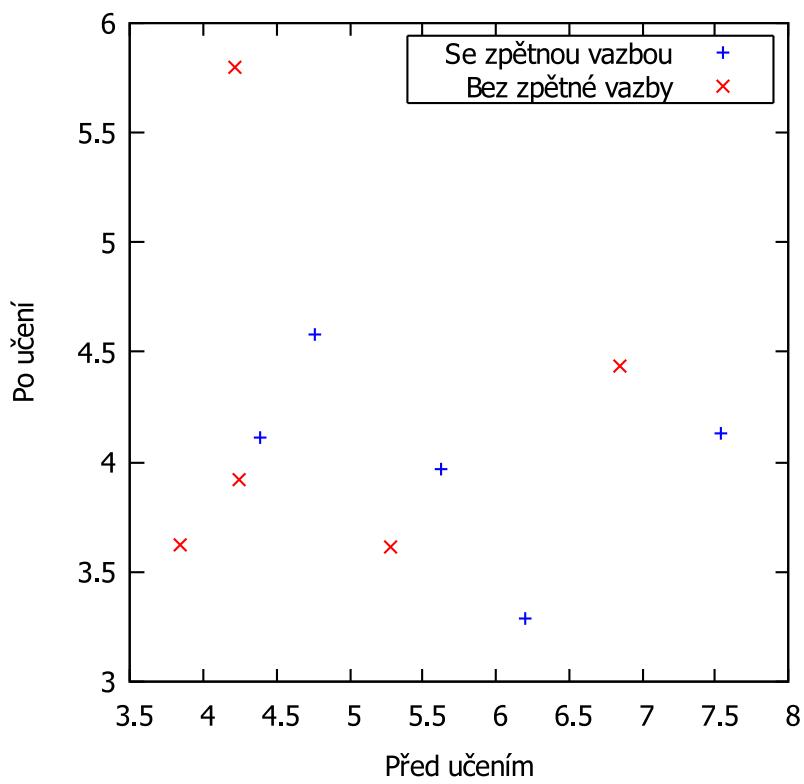
4. Měření

V této kapitole stručně představíme výsledky měření, detailní grafy ke každému pozorovateli zvlášt jsou v první příloze této práce.

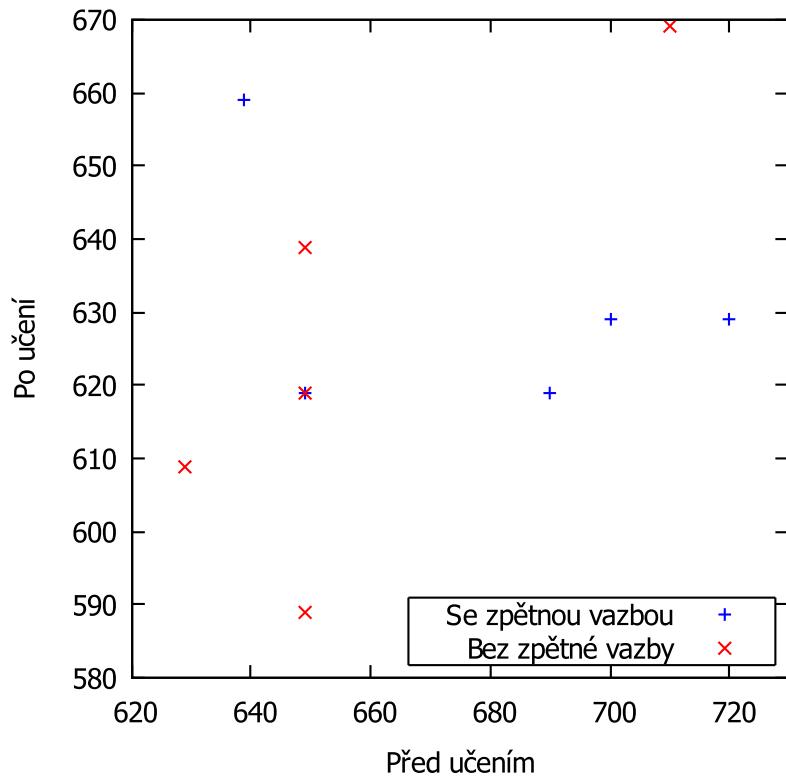
4.1 Výsledky



(a) Průměrný počet fixací



(a) Průměrný počet fixací v úspěšně splněných úkolech



(b) Obtížnost po 40. pokusu

5. Diskuse

Závěr

Seznam použité literatury

- BRADLEY, C., ABRAMS, J. a GEISLER, W. S. (2014). Retina-V1 model of detectability across the visual field. *Journal of Vision*, **14**(12), 22. doi: 10.1167/14.12.22. URL [+http://dx.doi.org/10.1167/14.12.22](http://dx.doi.org/10.1167/14.12.22).
- FOGEL, I. a SAGI, D. (1989). Gabor filters as texture discriminator. *Biological Cybernetics*, **61**(2), 103–113. ISSN 1432-0770. doi: 10.1007/BF00204594. URL <https://doi.org/10.1007/BF00204594>.
- GREEN, D. a SWETS, J. (1988). *Signal Detection Theory and Psychophysics*. Peninsula Pub. ISBN 9780932146236. URL <https://books.google.cz/books?id=AjWwQgAACAAJ>.
- HEEGER, D. (1997). Signal detection theory. URL <http://www.cns.nyu.edu/~david/handouts/sdt-advanced.pdf>.
- KINGDOM, F. A. A. a PRINS, N. (2009). *Psychophysics: A Practical Introduction*. Academic Press, 1 edition. ISBN 9780123736567.
- KOSKO, B. (2006). *Noise*. Viking. ISBN 9780670034956. URL <https://books.google.cz/books?id=skFGSAnR-0YC>.
- KRÁLÍČEK, P. (2002). *Úvod do speciální neurofyziologie*. Karolinum. ISBN 80-246-0350-0.
- MACMILLAN, N. A. a CREELMAN, C. D. (2004). *Detection theory: a user's guide*. Psychology Press, 2 edition. ISBN 0805842306, 9780805842302, 0805842314, 9780805842319.
- NAJEMNIK, J. a GEISLER, W. S. (2005). Optimal eye movement strategies in visual search. *Nature*, **434**, 387–391.
- NAJEMNIK, J. a GEISLER, W. S. (2008). Eye movement statistics in humans are consistent with an optimal search strategy. *Journal of Vision*, **8**(3), 4. doi: 10.1167/8.3.4. URL <http://dx.doi.org/10.1167/8.3.4>.
- NAJEMNIK, J. a GEISLER, W. S. (2009). Simple summation rule for optimal fixation selection in visual search. *Vision Research*, **49**(10), 1286 – 1294. ISSN 0042-6989. doi: <https://doi.org/10.1016/j.visres.2008.12.005>. URL <http://www.sciencedirect.com/science/article/pii/S0042698908006007>. Visual Attention: Psychophysics, electrophysiology and neuroimaging.
- PELI, E. (1990). Contrast in complex images. *JOSA A*, **7**(10), 2032–2040. doi: 10.1364/JOSAA.7.002032.
- PELLI, D. G. a FARELL, B. (1999). Why use noise? *JOSA A*, **16**(3), 647–653.
- POSNER, M. I. (1980). Orienting of attention. *Quarterly journal of experimental psychology*, **32**(1), 3–25.

- SHANNON, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, **27**(3), 379–423. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1948.tb01338.x.
- SMITH, P. L. a LITTLE, D. R. (2018). Small is beautiful: In defense of the small-n design. *Psychonomic Bulletin & Review*. ISSN 1531-5320. doi: 10.3758/s13423-018-1451-8. URL <https://doi.org/10.3758/s13423-018-1451-8>.
- SWETS, J. (1996). *Signal detection theory and ROC analysis in psychology and diagnostics: collected papers*. Scientific psychology series. Lawrence Erlbaum Associates. ISBN 9780805818345. URL <https://books.google.cz/books?id=1vkMAQAAQAAJ>.
- WOLFE, J. M., VÕ, M. L.-H., EVANS, K. K. a GREENE, M. R. (2011). Visual search in scenes involves selective and nonselective pathways. *Trends in cognitive sciences*, **15**(2), 77–84.

Problémy

Seznam obrázků

1.1	Závislost hit rate a false alarm rate pro různé hodnoty d' . Těmto křivkám se říká ROC křivky z anglického Receiver Operating Characteristic (Swets, 1996; Green a Swets, 1988).	7
1.2	Ukázky různých šumů. Tyto šumy byly vygenerovány pomocí algoritmu použitého přímo v aplikaci, která je součástí této práce. Algoritmus je blíže popsán ve druhé příloze.	8
1.3	Ukázky několika Gabor patchů. Všechny Gabor patche jsou 100 pixelů široké i vysoké. Levý patch má $\Theta = 1/4\pi$, ostatní mají $\Theta = -1/4\pi$, levý má jako obálku Gaussovou funkci, prostřední dva raised cosine, pravý lineární funkci vzdálenosti, první, druhý a čtvrtý patch mají frekvenci (v cyklech na pixel) 0.1, třetí 0.02. .	10
3.1	Příklady šumů s cíli.	17
3.2	Možné lokace Gabor patche	18

Rejstřík

C

Correct rejection, 4

D

d' , 6, 12, 36

d' mapa, 12, 19, 32

E

ELM pozorovatel, 2, 13, 35

Entropie, 10, 13

F

False alarm, 4

False alarm rate, 5

G

Gabor patch, 9, 11, 32

H

Hit, 4

Hit rate, 5

I

IBO, 13

Ideální Bayesovský pozorovatel, 2, 13

M

MAP pozorovatel, 13

Miss, 4

R

RMS kontrast, 16

S

Senzitivita, 5

Shannonova entropie, 10

Š

Šum, 7, 11

bílý, 8

červený, 8

modrý, 8

růžový, 8, 11, 36

Y

Y/N, 4

Yes-No problém, 4

Přílohy

První příloha – Detailnější výsledky měření

Druhá příloha – Dokumentace aplikace

V této příloze stručně popovídáme o aplikaci, která je součástí této práce.

Uživatelská dokumentace

Při spuštění aplikace se uživatel dostane na úvodní stránku, kde lze nastavit jméno účastníka, a lze z ní příslušnými tlačítky pokračovat k vlastnímu testování, do nastavení nebo na stránku s popisem úkolu, který účastníka experimentu čeká.

Na stránce s popisem úkolu si uživatel může přečíst, co ho čeká. V horní části obrazovky je vždy stručný popis části úkolu, a pod ní je šum, který tuto část úkolu ilustruje. K další části přejde kliknutím kamkoli na obrazovku. Délka ukázky se liší podle toho, zda má aktuální uživatel zapnutou zpětnou vazbu – v takovém případě jsou na konec přidány ještě čtyři části popisující zvuk. Po poslední instrukci je uživatel vrácen na obrazovku, ze které se na stránku s popisem úkolu dostal (tj. buď na úvodní stránku, nebo do nastavení).

Na stránce s nastavením je možné měnit mnoho různých parametrů vyhledávání. Pro replikaci experimentu tak, jak byl prováděn v této práci, stačí pouze nastavit jméno, zpětnou vazbu a v ostatních parametrech nechat defaultní hodnoty. Následující tabulka však nabízí popis toho, co přesně které nastavítko ovládá. U všech nastavitek, u nichž se ovládá velikost něčeho, se hodnota nastavuje v pixelech šumu.

Popisek	Význam
Difficulty	Průhlednost Gabor patche vynásobená tisícem a zaokrouhlená na jednotky. Editací tohoto pole se změní jak současná obtížnost, tak obtížnost, na kterou se hledání vrátí na začátku druhého a třetího testu. Hodnota v poli při vstupu do nastavení odpovídá první zmíněné obtížnosti.
Gabor patch diameter	Průměr obálky Gabor patche. Perioda se automaticky nastaví na třetinu této hodnoty.
Radius of uncovered area	V případě, že je vypnuto použití d' mapy, odkrývá se šum pomocí masky Raised cosine. Zde je možné nastavit poloměr masky v tomto případě.
Area uncovered for (ms)	Čas v milisekundách, na který se odkryje šum při doteчku.
Area uncovered for brief period only	Určuje, zda se šum odkrývá na trvalo (a jednotlivé dotečky se zobrazují současně), nebo pouze na dobu nastavenou v předchozím bodě. Při odkrývání natrvalo se však bez ohledu na nastavení vždy použije k odkrývání maska Raised cosine.

Popisek	Význam
Sound fully relative	Mění způsob, kterým se počítá výška zvuku. Při vypnutém nastavítku se používá vzorec uvedený v kapitole Metody. Při zapnutém se spočítá, kolik možných lokací cíle má z pohledu ELM vyšší hodnotu, než vybrané místo. Tón se pak vybere tak, aby jeho frekvence byla mezi 440 a 1760 Hertzů a byla exponenciálně závislá na tomto počtu.
Possible location distance	Vzdálenost možných lokací cíle
Regenerate noise	Návrat k měření se zapnutou touto možností způsobí kompletní restart současného úkolu, včetně přegenerování šumu a lokace Gabor patche. Vhodné zapnout, pokud jsme měnili některá jiná nastavení.
Preview	Přesune uživatele na stránku s popisem úkolu.
Start game	Zobrazí obrazovku s experimentem.
Back to main menu	Návrat na úvodní obrazovku.
Trials in first and third test	Počet úkolů v prvním a třetím testu.
Trials in second test	Počet úkolů během tréninku.
Correct/wrong responses in row before difficulty changes	Počet správných či špatných odpovědí v řadě, po kterých se změní obtížnost.
Difficulty changed by	O kolik se změní obtížnost (tisícem vynásobená průhlednost Gabor patche), když se má měnit (viz předchozí řádek).
Accuracy threshold	Maximální chyba v pixelech displeje, o kterou se uživatel může splést, aby byl úkol ještě uznán jako úspěšně splněný.
Maximal amount of fixations for successful trial	Maximální počet fixací, po nichž musí být označeno nalezení cíle, aby byl úkol považován za úspěšně splněný.

Popisek	Význam
Calculate score?	Vynuluje a zapne počítání a zobrazování skóre. Skóre získané jedním úkolem je 0, pokud cíl nebyl nalezen, jinak je určeno podle vzorce $\lfloor(1000 - (\text{Obtížnost})) / (\text{Počet fixací})\rfloor$.
d' map constants settings	Prvních šest nastavení v této kategorii určuje přímočárym způsobem konstanty d' mapy tak, jak jsou popsány v kapitole, která se jí věnuje. Hodnoty d'_0 a β je nutné zadávat vynásobené stem.
Recalculate d' map	Po změně parametrů d' mapy přepočítá pomocné datové struktury. Bez stlačení tohoto tlačítka by nemělo být možné po editaci konstant opustit nastavení. Přepočítávání se neděje automaticky po změně libovolné hodnoty, protože může trvat až několik vteřin a často je potřeba změnit více konstant současně.
Use d' map	Rozhoduje o tom, zda bude pro odkrývání šumu použita maska spočítaná z d' mapy, nebo Raised cosine.
Name	Identifikátor účastníka.
Feedback	Zapíná a vypíná zpětnou vazbu pro účastníka v druhém testu. V případě, že je jméno účastníka „unknown“, což je defaultní hodnota, je při zapnutí této funkce z ladících a předváděcích důvodů zpětná vazba poskytována bez ohledu to, v kterém testu se nachází.
Measure detectability	Spustí obrazovku, kde je uživateli prezentována řada pokusů, kdy se vždy Gabor patch se současnými parametry objeví ve středu s pravděpodobností 1/2. Uživatel pak může otestovat, jakou má při daných vlastnostech cíle hit rate a false alarm rate.
See log	Zobrazí log všech testů od posledního vymazání všech dat. Log zůstává napříč spuštěními aplikace a ukládá se automaticky po každém dokončeném úkolu.
Data deletion enabled	Je-li tato možnost aktivována, stlačení tlačítka „Delete all data“ opravdu smaže všechna data. V opačném případě je uživatel upozorněn na to, že musí mazání dat nejprve povolit.
Delete all data	Viz předchozí řádek.

Popisek	Význam
Generate noise	V případě, že je tato funkce vypnutá, bude místo růžového šumu vygenerována pouze šedá plocha.
Display locations	Je-li tato možnost zapnuta, bude v místě každé lokace zobrazen světlý kruh. Přesný odstín jeho pravé horní poloviny je závislý na aposteriorní pravděpodobnosti, kterou ELM pozorovatel přiřazuje právě této lokaci (čím světlejší půlkruh, tím vyšší pravděpodobnost). Odstín levé dolní poloviny je závislý na očekávaném snížení entropie, které podle ELM pozorovatele nastane, zafixujeme-li v dalším kroku tuto lokaci (opět světlejší kruh znamená větší očekávané snížení entropie). Ve středu kruhu, kde je očekávané snížení entropie největší (ve všech takových, není-li to jednoznačné) se nachází černá tečka.

V obrazovce, kde se provádí experiment, se vše koná tak, aby jediný nutný zásah do experimentu od administrátora byl, že bude sledovat počet uskutečněných úkolů (napsaný spolu s dalšími důležitými hodnotami v horní části obrazovky) a upozorní účastníka, když má experiment končit. Aplikace sama postupně mění obtížnost, dává zvukovou zpětnou vazbu v situacích, kdy má, a restartuje obtížnost na defaultní hodnotu na začátku druhého a třetího úkolu.

Programátorská dokumentace

Aplikace je napsána v jazyce Swift, který je určený pro psaní nativně běžících aplikací pro operační systém iOS. Je určená na zařízení iPad Air.

V této sekci stručně přiblížíme dva netriviální algoritmy, které se v aplikaci vyskytují. Bude se jednat o výpočet očekávaného snížení entropie v rámci simulace ELM pozorovatele, a generování růžového šumu.

V obou těchto algoritmech je potřeba generovat náhodná čísla s normálním rozdělením. Na to použijeme funkci `GKGaussianDistribution.nextUniform()`, která je součástí knihovny `GameplayKit`, jedné ze standardních knihoven jazyka Swift. Tato funkce se od normálního rozdělení odklání v tom, že nepřipouští hodnoty, které se od střední hodnoty rozdělení liší o více než tři standardní odchylky. Protože je však pravděpodobnost takových hodnot v normálním rozdělení rovna přibližně 0.003, rozhodli jsme se, že ovlivnění experimentu touto odlišností zanedbáme.

Model ELM pozorovatele

ELM byl v práci vytvořen přesně tak, jak byla jeho simulace popsána v článku (Najemnik a Geisler, 2009).

To znamená, že při fixaci bodu x dostane z každé lokace y jako pozorování reálné číslo, které je náhodně vybrané z normálního rozdělení, jehož střední hodnota je -0.5 , pokud se cíl v daném bodě nenachází, a 0.5 pokud ano. Směrodatná

odchylka tohoto rozdělení je rovna hodnotě $d'(y - x)$. Poté spočítá veškerou informaci, kterou již o jednotlivých lokacích dostal, jako součet odpovědí, které dostal při předchozích fixacích, vynásobený druhými mocninami příslušných hodnot d' (ve skutečnosti stačí přičíst odpovědi z aktuální fixace k již spočítanému součtu po předchozí fixaci).

V dalším kroku spočítá aposteriorní pravděpodobnosti prostě tak že pro získaný součet s nastaví aposteriorní pravděpodobnost na e^s , a pak ještě všechny aposteriorní pravděpodobnosti vydělí jejich součtem, aby byl po vydělení součet roven jedné. Až po sem je algoritmus lineární v počtu možných lokací. Nyní je však potřeba spočítat očekávané snížení entropie. To se spočítá podle již dříve uvedeného vzorce jako

$$\sum_{i \in L} p_T(i) d'^2(i - k(T + 1)),$$

kde L je množina možných lokací, $p_T(i)$ je aposteriorní pravděpodobnost, že se cíl nachází v lokaci i a $k(T + 1)$ je lokace, jejíž očekávané snížení entropie počítáme.

Před první fixací se očekávaná snížení entropie počítají obdobně, až na to, že jako pole s předchozími odpovědi použijeme pole vyplněné samými nulami.

Důkaz korektnosti algoritmu zde nebudeme uvádět, neboť je již uvedený v článcích (Najemník a Geisler, 2005, 2008, 2009).

Generování růžového šumu

Šum generujeme tak, že nejprve vygenerujeme dvojrozměrné pole náhodných komplexních čísel stejně velké, jak velký chceme výsledný šum. Tato komplexní čísla mají v goniometrické podobě rovnoměrně náhodné argumenty. Absolutní hodnotu čísla na pozici (x,y) v poli $n \times n$ vybereme z normálního rozdělení se střední hodnotou

$$\left(\min(x, n-x)^2 + \min(y, n-y)^2 \right)^{-2} \quad (1)$$

a směrodatnou odchylkou rovnou třetině této hodnoty. V případě, kdy by tato hodnota vyšla jako nekonečno, můžeme hodnotu prostě zahodit (zdůvodnění tohoto tvrzení dále vyplýne z algoritmu) a použít libovolnou. Výběr střední hodnoty vychází z definice růžového šumu, odchylka byla zvolena při testování kvůli výše uvedeným vlastnostem náhodného generátoru (nemůže se nám stát, že by vyšla záporná absolutní hodnota), a byla na této hodnotě ponechána, protože algoritmus dával vizuálně žádoucí výsledky a definice růžového šumu ji nespecifikuje.

Na takto vytvořené pole zavoláme (zpětnou, ale ve skutečnosti na tom nezáleží) dvojrozměrnou rychlou Fourierovu transformaci¹. Výsledkem je růžový šum. My ho ale potřebujeme být schopni zobrazit. Nejprve tedy zahodíme imaginární složku komplexních čísel. Poté spočítáme jeho průměr a směrodatnou odchylku a uplatníme na něj lineární transformaci takovou, aby jeho střední hodnota byla 128 a směrodatná odchylka 64. Nakonec záporné hodnoty nastavíme na nulu a hodnoty převyšující 255 na 255. Nakonec nastavíme na 128 ty pixely, jejichž (euklidovská) vzdálenost od středu obrazu je větší, než $n/2$, čímž dosáhneme toho, že výsledné pole bude kruhové. Výsledné pole interpretujeme jako černobílý obraz s osmi bity na pixel.

¹Dvojrozměrná diskrétní Fourierova transformace je Fourierova transformace nejprve provedená po řádcích a potom po sloupcích.

Fourierovu transformaci nalezneme v knihovně **Accelerate**, která je jednou ze standardních knihoven jazyka Swift.

Nyní se vrátíme k tomu, proč jsme mohli zahodit hodnotu, pro níž vyšel nekonečný koeficient. Ten vyjde právě tehdy, když jsou ve vzorci (1) obě sčítané závorky nulové. Budeme-li x i y indexovat od nuly, znamená to, že musí být x i y rovno nule. To znamená, že se tato situace týká jednoho jediného bodu, a to bodu $(0,0)$. Protože ale poté voláme Fourierovu transformaci, z nultého bodu se stane hodnota, která se násobí pouze jedničkou, tedy ji bez změny přičteme ke každé hodnotě. Protože ale potom stejně šum lineárně transformujeme, aby měl správnou střední hodnotu a směrodatnou odchylku, žádná předchozí nekonstantní lineární transformace výstup neovlivní (nebo alespoň ne více, než tím, že způsobí nějaké nechtěné zaokrouhllování², to ale nebude mít v praxi významný vliv).

Pro generování jiného z šumů, o kterých se mluví v sekci o barvách šumu, stačí ve vzorci (1) vynásobit hodnotu -2 v exponentu příslušnou hodnotou β .

²V algoritmu je komplexní číslo reprezentováno pomocí dvou hodnot typu **Double**.