
Tárgykövető algoritmusok matematikai modellekkel

XI. Erdélyi Tudományos Diákköri Konferencia – Kolozsvár, 2008. május 23-24

Szerző:

BÓCSI BOTOND ATTILA

BABES-BOLYAI TUDOMÁNYEGYETEM

MATEMATIKA ÉS INFORMATIKA KAR

INFORMATIKA SZAK, IV. ÉV

Témavezető:

DR. CSATÓ LEHEL

ADJUNKTUS

BABES-BOLYAI TUDOMÁNYEGYETEM

MATEMATIKA ÉS INFORMATIKA KAR

PROGRAMOZÁSI NYELVEK ÉS MÓDSZEREK

TANSZÉK

Kivonat

A dolgozat témája tárgykövető algoritmusok összehasonlító elemzésének a megvalósítása, úgynevezett szűrő módszerek (*filtering methods*) felhasználásával. Bemutatásra kerülnek az algoritmusok alapjául szolgáló matematikai modellek és annak módja, hogy hogyan lehet alkalmazni őket a tárgykövetés területén. A dolgozat célja egy objektív összehasonlítás megvalósítása, amely számos tulajdonságra kiterjed: sebesség, pontosság, robusztusság, stb. Az összehasonlítás figyelembe veszi a környezeti tényezőket is.

Az elemzés a tárgykövető algoritmusok egy speciális családjára terjed ki, az úgynevezett szűrő módszerekre. A következő három algoritmus mutatom be: a Kálmán szűrő (Kalman filter), valamint két nemlineáris kiterjesztése: az "unscented" Kálmán szűrő (unscented Kalman filter) és a részecskeszűrő (particle filter).

Kulcsszavak: tárgykövetés, unscented Kálmán szűrő, részecskeszűrő

1. Bevezetés

A modern élet számos területén elengedhetlenné vált a hatékony videófeldolgozás és ezen belül a hatékony tárgykövető algoritmusok használata. Majdnem minden robot, biztonsági kamera és ipari kamera számára nélkülözhetetlen feladat különböző tárgyak felismerése és követése. Számos módszer létezik, ami a hatékony tárgykövetést hivatott megvalósítani, de mindegyiknek megvannak a saját gyengéi és erősségei. Ezen gyengék és erősségek között választani kell, főleg abban az esetben, amikor a követést megvalósító hardver nem teszi lehetővé a sok számolást igénylő módszerek használatát. Ennek a megszorításnak legtöbbször az az ára, hogy kevésbé pontos követést kapunk. Ezen felül, a megfelelő algoritmus kiválasztásánál lényeges szempont kell legyen a környezet, amelyben használni szeretnénk. A megvilágítás, a háttér, a tárgy mozgásának típusa más és más hatással lehet az egyes módszerek teljesítményére.

Egy speciális algoritmus osztály néhány tagját mutatom be: a *szűrőket*. A szűrő elnevezés a fizika tudományából származik, ahol egy olyan szerkezetet jelöl, ami a nem kívánt összetevőket szűri ki valamilyen jelsorozatból (például elektromos jelekből). A megnevezés a becslélméletbe az 1940-es években került be [Grewal és Andrews, 2001],

ahol azokat a módszereket foglalta magában, amelyeket a dinamikus rendszerekben megjelenő zaj kiszűrésére használtak.

Fontos megjegyezni, hogy lényeges különbség van az egyes objektumok felismerése és követése között. Azonosítani egy tárgyat egy zajos környezetben inkább ontológia kérdés, ami az ittenitől eltérő megközelítést kíván meg. Ebben a dolgozatban nem foglalkozom a tárgyfelismerés kérdésével, inkább a már azonosított objektumok követése a cél.

A szűrő módszerek mellett létezik egy másik fontos tárgykövető algoritmus kategória, a *condensation* algoritmus, amit Isard és Blake [1996, 1998] használt először 1996-ban. A módszer lényeges hátulütője, hogy nem használható tetszőleges tárgyábrázolás mellett, szükségszerűen kontúr alapú ábrázolást követel meg, ami sokszor nagy számítási igénnyel jár. Más objektumábrázolási módszerekkel később foglalkozom.

A dolgozat eleje egy rövid bevezetőt tartalmaz a dinamikus rendszerek témakörbe. Ezt követi a tárgykövető algoritmusok alapjául szolgáló matematikai modellek részletes leírása, valamint az elért kísérleti eredmények bemutatása, elemzése és magyarázása.

2. Tárgykövető algoritmusok

A tárgykövető algoritmusok alapját képező matematikai modellek megértéséhez elengedhetetlen a dinamikus rendszerek [Haykin, 2001] fogalmának a megértése. Egy dinamikus rendszer olyan elemek összessége, amelyek tulajdonságai időben változnak. Minden időpillanatban a rendszer állapota az előző állapotoktól függ. Amennyiben a rendszer az n előző állapottól függ, leírható egy n -ed rendű differenciálegyenlettel. Nagyon gyakran a rendszer állapotának a változását nem lehet explicit módon megadni. A dinamikus rendszerek legnagyobb problémája az, hogy nagyon nehezen lehet az időbeli változásokukat előrejelezni, egy kis változás a kezdeti feltételekben óriási hatással bírhat a jövőbeli állapotokra nézve.

Ezen felül, szintén nehéz feladat - néha lehetetlen - a rendszer állapotának az explicit módon való mérése és csak közvetett adatok állnak rendelkezésre.

Az is előfordulhat, hogy az általunk használt mérőműszer nem képes tetszőlegesen pontos adatokat szolgáltatni a rendszer kimenetére vonatkozóan, vagyis a mérések zajosak

(mérési zaj). Az is lehet zaj forrása, hogy nem ismerjük pontosan a rendszer fejlődését irányító szabályokat (működési zaj).

Amennyiben a tárgykövetés feladatára úgy tekintünk, mint egy dinamikus rendszer előrejelzésére, matematikailag bizonyított módszereket tudunk felhasználni. A követendő tárgy pozíciója játssza a rendszer belső állapotának a szerepét, míg a kamerától kapott kép a mérési adatot.

A továbbiakban a probléma három matematikai megközelítését mutatom be, amelyeket sikerrel alkalmaznak a tárgykövetés területén.

2.1. Kálmán szűrő

A Kálmán szűrő lineáris dinamikus rendszerek optimális becslését teszi lehetővé, amennyiben zajos adatok állnak rendelkezésre és úgy a működési, mint a mérési zaj Gauss eloszlású. 1960-ban Rudolf Kálmán fejlesztette ki [Kalman, 1960]. Először 1969-ben alkalmazták, amikor a NASA Apollo program keretein belül a pályaszámításban használták fel [Grewal és Andrews, 2001].

A témakörbe egy rövid, ugyanakkor tiszta és világos bevezetést Welch és Bishop [1995] munkája tartalmaz. Részletesebb leírást Grewal és Andrews [2001], illetve Haykin [2001] könyveiben találhatunk. Ezek tartalmazzák a szűrő működését leíró egyenletek szigorú levezetését.

A Kálmán szűrő által használt állapotér a következő képen írható fel:

$$\begin{aligned}x_k &= \mathbf{F}_k x_{k-1} + \mathbf{B}_k u_k + w_k \\z_k &= \mathbf{H}_k x_k + v_k,\end{aligned}\tag{1}$$

ahol x_k a rendszer belső állapotát jelenti; z_k a mérési adat; u_k a kontroll input információ; \mathbf{F}_k a $k - 1$ és a k időpillanatok közti áttérési mátrix; \mathbf{H}_k a mérési adatok és a belső állapot kapcsolata. Amennyiben feltételezzük, hogy a kontroll információn keresztül beavatkozhatunk a rendszerbe, akkor ezt a \mathbf{B}_k írja le. w_k és v_k a működési és mérési zajt jelentik. Mindkét zaj szükségszerűen *additív fehér zaj*¹. Egy másik lényeges megszorítás a zajok természetét illetően az, hogy ezek nulla várható értékűek és Gauss eloszlásúak

¹A fehér zaj olyan véletlen jelsorozat, amely egyenletes spektrális eloszlással rendelkezik.

kell legyenek. Valós alkalmazásokban (a tárgykövetésben) feltételezhetjük, hogy a rendszer stacionárius, vagyis a \mathbf{F} , \mathbf{H} , \mathbf{B} , w és v értékek nem függenek az időtől, így a k index elhagyható.

A szűrő működését két lépésre lehet felosztani:

1 az első lépés a *jóslás*, amikor a becslés kizárólag az előző állapotra (x_{k-1}) támaszkodik:

$$x_k^- = \mathbf{F}_k x_{k-1}^+ + \mathbf{B} u_k \quad (2)$$

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1}^+ \mathbf{F}_k^T + \mathbf{Q}, \quad (3)$$

ahol az első sor az optimális becslés, ismelve az előző állapotot (x_{k-1}^+) és a kontroll input információt. A második sor a belső állapot kovarianciáját határozza meg, ismelve a előző állapot és a működési zaj kovarianciáját.

2 a második lépés a *frissítés*, amikor az új mérési eredmények is rendelkezésre állnak:

$$x_k^+ = x_k^- + \mathbf{K}_k (z_k - \mathbf{H}_k x_k^-) \quad (4)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (5)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1}, \quad (6)$$

ahol x_k^+ a belső állapotra kapott végső becslés. Ennek értéke az előző lépés eredményén (2) és a mérési adaton alapszik. Az \mathbf{R} a mérési zaj kovarianciáját leíró mátrix. A \mathbf{K}_k -t Kálmán nyereségnek nevezzük (*Kalman gain*), ez határozza meg, hogy mekkora hatása van a mérési adatnak a becslésre.

Érdemes megjegyezni, hogy a szűrő által generált becslések kizárólag az előző lépések állapotaitól és az aktuális mérési eredményektől függenek. Gyakorlati szempontból ez azt jelenti, hogy a hatékony működéshez kevés információ tárolása szükséges.

Egy közelebbi pillantást vetve a (3) egyenletre, megfigyelhetjük a működése zaj természetének a szűrő működésére gyakorolt hatását. Növelve a \mathbf{Q} értékét, \mathbf{P}_k^- szintén növekszik, így az előző állapot kevésbé válik fontossá. A \mathbf{K}_k Kálmán nyereség a \mathbf{R} -tól függ. A (6) egyenletből kitűnik, hogy amennyiben \mathbf{R} végtelenhez tart (nagyon bizonytalanok a méréseink), \mathbf{K}_k nullához közelít, vagyis az mérési adat nem befolyásolja a becsült értéket. A másik végletet tekintve, ha \mathbf{R} nullához tart, akkor \mathbf{K}_k értéke \mathbf{H}_k^{-1} -hoz közelít, így a jóslás eredménye lesz elhanyagolva.

2.2. Unscented Kálmán szűrő

A Kálmán szűrő legnagyobb hátránya az állapotter linearitásának és a zaj természetének a szükségességében rejlik. Az unscented Kálmán szűrő ezen megszorításokat hivatott kiküszöbölni statisztikai megközelítést használva. Az első publikáció a szűrőről 1995-ből származik és Uhlmann [1995] nevéhez kapcsolódik. Részletesebb leírást Simon és Uhlmann [1997] munkájában találhatunk.

A megközelítés alapja az az intuíció, hogy könnyebb megbecsülni egy valószínűségi eloszlást, mint egy nemlineáris függvényt. Ennek eléréséhez úgynevezett *szigma pontokat* vezetnek be, amivel a rejtett állapot szórását (\mathbf{P}_k) próbálták jellemezni. Ezen pontok kiválasztásánál figyelembe kell venni, hogy kellő pontossággal írják le a rendszer első és második momentumát (várható értékét és szórását), feltételezve, hogy a belső állapot Gauss eloszlású. Az alapötlet az, hogy a szigma pontokon végezzük el a transzformációt – aminek nem kell lineárisnak lennie – és az így keletkezett pontokat felhasználva becsüljük a rendszer állapotát és szórását (kovarianciáját).

Bizonyított [Simon és Uhlmann, 1997], hogy ezt a módszert használva a nemlineáris transzformációk magasabb rendű tagjait (Tylor sorbafejtés esetén) sem hagyjuk figyelmen kívül.

Amint azt láttuk a módszer által használt állapotter nem feltétlenül lineáris:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_k$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k,$$

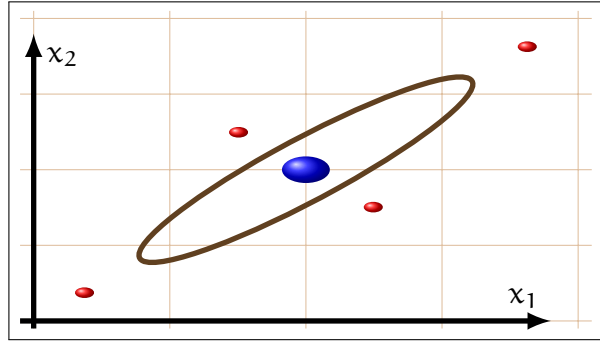
ahol $f()$ és $h()$ tetszőleges lineáris vagy nemlineáris függvények, \mathbf{w}_k és \mathbf{v}_k Gauss eloszlású fehér zajok. Az unscented Kálmán szűrő leglényegesebb eleme a szigma pontok kiválasztása, a továbbiakban \mathcal{X} -val jelöljük őket és a meghatározásuk a következő egyenleteken alapszik:

$$\mathcal{X}_0 = \mathbf{x}_k$$

$$\mathcal{X}_i = \mathbf{x}_k + \left(\sqrt{(n + \lambda) \mathbf{P}_x} \right)_i \quad i = \overline{1, n} \quad (7)$$

$$\mathcal{X}_{n-i+1} = \mathbf{x}_k - \left(\sqrt{(n + \lambda) \mathbf{P}_x} \right)_{n-i+1}, \quad (8)$$

ahol $(\dots)_i$ a zárójelben lévő mátrix i -edik oszlopát jelenti; \mathcal{X} a pontokat tartalmazó hal-



1. ábra. Szigma pontok

maz és

$$\lambda = \alpha^2(n + \kappa) - n,$$

ahol α és κ skálázó paraméterek, melyek plusz szabadságfokokat kölcsönöznek a modellnek. Az α -n keresztül a rendszer szórását lehet alá- illetve túlbecsülni, ami a szigma pontok szétszórtságát határozza meg a eredeti pozícióhoz képest. Általában ezen paraméterek kicsi értékkel rendelkeznek, $\alpha = 10^{-3}$ és $\kappa = 0$ (lásd Simon és Uhlmann [1997]).

Az 1. ábrán egy illusztrációt láthatunk, ami a szigma pontok elhelyezkedését mutatja be kétdimenziós esetben. A nagy – kék – pont az eredeti állapotot jelképezi, az ellipszis a kovarianciát és a kicsi – piros – pontok a generált szigma pontokat, amihez még hozzá kell venni magát az eredeti pontot is.

A (7) és (8) egyenletekben egy mátrix gyökét kell kiszámolni. Ez egy \mathbf{B} mátrixot jelent, úgy, hogy $\mathbf{P}_\kappa = \mathbf{B}\mathbf{B}^\top$. A \mathbf{P}_κ mátrixnak mindig létezik az ilyen formájú dekompozíciója [Press et al., 1992] és meghatározásához ajánlott a Cholesky dekompozíció [Simon és Uhlmann, 1997].

Minden szigma pont rendelkezik egy súllyal, amit a következő képpen számolhatunk ki:

$$W_0^{(m)} = \frac{\lambda}{n + \lambda} \quad (9)$$

$$W_0^{(c)} = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta) \quad i = \overline{1, 2n} \quad (10)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n + \lambda)}, \quad (11)$$

ahol β a belső állapot eloszlásáról rendelkezésünkre álló a-priori információt testesíti meg. Gauss eloszlás esetén a 2-es érték a legmegfelelőbb [van der Merwe et al., 2000].

Egy n dimenziós állapot esetén $2n + 1$ szigma pontot generálunk.

Matematikailag a szűrő működését a következő képpen lehet megfogalmazni:

- **jóslás** (time update):

$$\begin{aligned}\mathcal{X}_{k|k-1} &= f(\mathcal{X}_{k-1}) \\ \hat{x}_k^- &= \sum_{i=0}^{2n} W_i^{(m)}(\mathcal{X}_{k|k-1})_i \\ \mathbf{P}_k^- &= \sum_{i=0}^{2n} W_i^{(c)}[\mathcal{X}_{k|k-1} - \hat{x}_k^-][\mathcal{X}_{k|k-1} - \hat{x}_k^-]^T \\ \mathcal{Z}_{k|k-1} &= h(\mathcal{X}_{k|k-1}) \\ \hat{z}_k^- &= \sum_{i=0}^{2n} W_i^{(m)}(\mathcal{Z}_{k|k-1})_i\end{aligned}$$

- **frissítés** (measurement update):

$$\begin{aligned}\mathbf{P}_{\hat{z}_k, \hat{z}_k} &= \sum_{i=0}^{2n} W_i^{(c)}[\mathcal{Z}_{k|k-1} - \hat{z}_k^-][\mathcal{Z}_{k|k-1} - \hat{z}_k^-]^T \\ \mathbf{P}_{\hat{x}_k, \hat{z}_k} &= \sum_{i=0}^{2n} W_i^{(c)}[\mathcal{X}_{k|k-1} - \hat{x}_k^-][\mathcal{Z}_{k|k-1} - \hat{z}_k^-]^T \\ \mathbf{K} &= \mathbf{P}_{\hat{x}_k, \hat{z}_k} \mathbf{P}_{\hat{z}_k, \hat{z}_k}^{-1} \\ x_k &= \hat{x}_k^- + \mathbf{K}(z_k - \hat{z}_k) \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K} \mathbf{P}_{\hat{z}_k, \hat{z}_k} \mathbf{K}^T,\end{aligned}$$

ahol \hat{x}_k^- és \hat{z}_k^- a belső állapotot, illetve a mérési adatot jelentik; \mathbf{P}_k^- a \hat{x}_k^- a-priori kovarianciáját jelöli; $\mathbf{P}_{\hat{z}_k, \hat{z}_k}$ és $\mathbf{P}_{\hat{x}_k, \hat{z}_k}$ empirikus korrelációs mátrixok.

A \mathbf{K} mátrix a (6) egyenletben szereplő Kálmán nyereségnek feleltethető meg, értéke a korrelációs mátrixok értékeitől függ: növelve a \hat{z}_k^- szórását, az mérési adat (z_k) hatása csökken a becslésben. A \hat{x}_k^- és \hat{z}_k^- közötti korreláció fordított hatással bír.

2.3. Részecskeszűrő

A részecskeszűrő egy hatékony algoritmus ismeretlen valószínűségi eloszlások szimulálás útján való közelítésére. A módszer a már bemutatott algoritmusok kiterjesztése olyan esetre, amikor a rendszer nemlineáris és a zajról sem feltételezzük, hogy Gauss eloszlású.

Kifejlesztése 1993-ra tehető és Gordon et al. [1993] nevéhez fűződik. Részletes leírását megtaláljuk Fox [2001] és Robert és Casella [2004] munkáiban.

Az alapötlet az, hogy egy tetszőleges valószínűségi eloszlást definiálni lehet egy súlyozott ponthalmazzal. Ez a halmaz $\langle x_k^{(i)}, w_k^{(i)} \rangle$ számpárokból áll, ahol $x_k^{(i)}$ a rendszer egy lehetséges állapotát jelenti, míg $w_k^{(i)}$ ennek az állapotnak a valószínűségét. A szűrő használatakor egy véletlen súlyozott ponthalmazt generálunk, amivel megpróbáljuk leírni a rendszer a-priori eloszlását. Az a-poszteriori eloszlás meghatározásához egy új ponthalmazt hozunk létre, ami a $p(x_k|x_{k-1})$ -ből való mintavételezésből származik. Ezzel leírható a rendszer dinamikája. A mérési adatok hatásai a pontok súlyain keresztül érvényesülnek, vagyis minden pont olyan súlyt kap, ami az illető pont valószínűségét fejezi ki. Ez a $p(z_k|x_k)$ feltételes valószínűséggel adható meg.

Amennyiben a részecskeszűrőt az előbb leírt formában próbálnánk implementálni, a működése instabil lenne. Ez abban nyilvánul meg, hogy egy idő után minden súly nullához közeli értéket kap, kivéve egyet, ami egyhez tart. Ekkor használhatatlanná válik a szűrő. Ennek a jelenségnek az elkerüléséhez ajánlott, hogy az a-priori ponthalmazból visszatevéssel válasszunk pontokat, azzal a valószínűséggel, amit az illető pont súlya megad [Doucet et al., 2005].

Egy másik lényeges kérdés a ponthalmaz méretének oly módon való megválasztása, hogy a leghatékonyabb becslést tudjunk adni az a-poszteriori eloszlásra. Kong et al. [1994] szerint az optimális ponthalmazméretet az pontok súlyaiból kaphatjuk meg a következő képpen:

$$N_{\text{eff}} = \frac{N_s}{1 + \text{var}(w_k)},$$

ahol N_s az eredeti ponthalmaz mérete.

Látszik, hogy minden lépésben $N_{\text{eff}} \leq N_s$, vagyis a halmaz mérete időben csökken. Ahhoz, hogy megállítsuk ezt a csökkenést, egy határt kell bevezetni, ami alatt visszaállunk egy nagyobb ponthalmazméretre.

A módszer működését a 1. algoritmus tartalmazza. Minden lépésben az a cél, hogy megbecsüljük az a-poszteriori eloszlást a S_k ponthalmazzal, tudva, hogy az a-priori eloszlás a S_{k-1} ponthalmaz definiálja – 1. sor. A visszatevéssel való kiválasztás a 5-12. sorokban történik. Először egy j indexet határozunk meg $w_{k-1}^{(j)}$ valószínűséggel –

Algorithm 1 Részecskeszűrő

```
1: Input:  $S_{k-1} = \{\langle x_{k-1}^{(i)}, w_{k-1}^{(i)} \rangle | i = \overline{1, n}\}$ 
2:
3:  $S_k \leftarrow \emptyset$ 
4:  $\alpha \leftarrow 0$ 
5: for  $i \leftarrow \overline{1, n}$  do
6:    $j \sim p(j|w_{k-1})$ 
7:    $x_k^{(i)} \sim p(x_k | x_{k-1} = x_{k-1}^{(j)})$ 
8:    $w_k^{(i)} \sim p(z_k | x_k = x_k^{(j)})$ 
9:
10:   $\alpha \leftarrow \alpha + w_k^{(i)}$ 
11:   $S_k \leftarrow S_k \cup \{\langle x_k^{(i)}, w_k^{(i)} \rangle\}$ 
12: end for
13:
14: for  $i \leftarrow \overline{1, n}$  do
15:   $w_k^{(i)} \leftarrow w_k^{(i)} / \alpha$ 
16: end for
17:
18: return  $S_k$ 
```

6. sor². Az index meghatározása után felhasználjuk a rendszer dinamikájáról rendelkezésünkre álló információkat – 7. sor. Ez a frissítés lépésnek felel meg a Kálmán szűrők esetén.

A mérési eredmény közvetlen módon nem befolyásolja a pontokat, hatása a súlyokon keresztül érvényesül, annak függvényében, hogy mennyire valószínű a mérési érték, feltételezve, hogy a rendszer az illető állapotban van – 8. sor. Ahhoz, hogy a súlyok egy érvényes valószínűségi eloszlást definiáljanak, összegük egy kell legyen, vagyis az értékeket normalizálni kell – 14-16. sor.

Mielőtt áttérnék a kísérleti eredmények bemutatására, megemlítenék egy harmadik módszert a nemlineáris dinamikus rendszerek előrejelzésére: a kiterjesztett Kálmán szűrő (extended Kalman filter) vagy EKF [Ribeiro, 2004]. A szűrő a nemlinearitást a függvény linearizációja útján küszöböli ki. Ehhez a Taylor sorbafejtést használja fel és a sorbafejtésből mindössze az elsőrendű tagokat veszi figyelembe. A módszer használatá-

²Tárolva a kumulatív valószínűségeket a visszatevéssel való kiválasztás $\mathcal{O}(n)$ időben megvalósítható.

nak számos hátulütője van: a Jacobi mátrixok kiszámítása időigényes lehet bonyolult függvények esetében; az ilyen típusú linearizáció nagyarányú hibákat eredményezhet, ha a függvény *nagyon* nemlineáris és a Tylor sorbafejtés magasabb rendű tagjai is fontossá válnak. A módszerrel a továbbiakban nem foglalkozom, mivel valós alkalmazások esetén nem használható. A linearizációt általában nem lehet megvalósítani, mivel a tárgy mozgását leíró függvényről semmit nem tudunk.

3. Eredmények

Sok esetben a követendő tárgy ábrázolása és a követését végző algoritmus nem választható külön, annak ellenére, hogy ezek függetlenek kellene legyenek. Például a *condensation* algoritmus (1. fejezet) csak kontúr alapú reprezentációt tesz lehetővé. A dolgozatban hisztogramot használó objektumábrázolást használtam. A hisztogram alapú ábrázolás lehetővé teszi, hogy tetszőleges tárgyat tudjunk követni, anélkül, hogy ez lényeges erőforrásokat vonna el.

A hisztogram alapú ábrázolást használva a kép egy 3×256 dimenziós tér egy eleme, ahol a 3 a kép három színekomponeusét (piros, zöld, kék) jelöli, a 256 pedig a színekomponeus intenzitásainak a lehetséges értékei.

Ha ezt az ábrázolási módot használom a kép térbeli struktúrája elveszlik, de egy megfelelő objektumot választva – például egy egyszínű labdát – ez nem befolyásolja a rendszer működését és az ábrázolás alacsony számítási igényvel rendelkezik. Ez akkor fontos, amikor hisztogramok távolságát kell kiszámolni.³

Ahhoz hogy számszerűsíteni lehessen az algoritmusok hatékonyságát, egy keretrendszer dolgoztam ki, ami abból állt, hogy egy monokróm labda (korong) mozog a képernyőn egy téglalap formájú területen. Az algoritmusnak ezt a labdát kell követnie, úgy, hogy semmilyen információval nem rendelkezik annak mozgásáról. A labda és az asztal színeit (fekete és fehér) úgy választottam meg, hogy kontrasztosak legyenek, ezzel is segítve a felismerést. A kísérlet során igyekeztem minimalizálni a környezeti hatásokat, de elkerülhetetlen, hogy a fényviszonyok ne befolyásolják a méréseket.

³Hisztogramok távolságának a kiszámításához L_2 normát használtam.

A labda különböző mozgástípusokkal rendelkezhet: lineáris, nemlineáris⁴, különböző sebességgel mozoghat és a mozgáshoz változtatható mértékű additív zajt lehet hozzáadni. Negyven kísérletet végeztem, úgy, hogy mindegyik tíz percet tartott. Egy 200×160 kamerát használtam, amit 22 kép per másodperc sebességre volt képes.

Öt algoritmust teszteltem: a Kálmán szűrőt (KF), az unscented Kálmán szűrőt (UKF), a részecskeszűrőt 10 szigma ponttal (PF₁₀), a részecskeszűrőt 30 szigma ponttal (PF₃₀) és a részecskeszűrőt 70 szigma ponttal (PF₇₀). A kísérletek során a következő tulajdonságokat mértem: (1) kép per másodperc, vagyis, hogy milyen intenzitással kéri az algoritmus a képkockákat a kamerától (ez a számítási igényről árulkodik); (2) annak a száma, hogy hányszor veszítette el az algoritmus a labdát; (3) mennyi idő telt el, amíg újra megtalálta a labdát – explicit tárgykeresést nem használtam.

A 1. táblázat néhány általános információt tartalmaz az algoritmusokról: *fps* - kép per másodperc (frame per second); *hiba* - a labda valós pozíciója és a becsült pozíció közti távolság; *elveszít* - hányszor veszítette el az algoritmus a labdát; *talál* - mennyi idő telik el a labda megtalálásáig.

1. táblázat. Algoritmusok teljesítménye

Alg.	fps	hiba	elveszít	talál
PF ₁₀	15.81	11.31	1.25	6.25
PF ₃₀	10.71	12.28	11.75	5.67
PF ₇₀	7.24	15.69	18.25	4.41
KF	10.28	8.57	7.5	8.02
UKF	10.19	8.52	4.0	8.87

Figyelembe véve a módszerek sebességét, a részecskeszűrő egy extra szabadságfokkal rendelkezik, mivel skálázni lehet a felhasznált ponthalmaz mérete szerint. Ettől függően lehet gyorsabb vagy lassabb, mint a KF vagy az UKF. Látszik, hogy a KF és az UKF közel hasonló sebességre képes, úgy, hogy az UKF pontosabb becsléseket ad. Azt a kritériumot figyelembe véve, hogy hányszor veszíti el a labdát az algoritmus a PF₁₀ meglepően jó eredményeket produkált, ezt az UKF és a KF követi.

⁴Szigorú értelemben véve egyik felhasznált mozgástípus sem lineáris, mivel a labda visszaverődik a szimulációs környezet széléről.

Egy másik fontos mérőszám, az hogy az algoritmus mennyi idő után találja meg a labdát, miután elveszítette. A KF és az UKF nem rendelkezik implicit kereső mechanizmussal, így ebben a tekintetben ezek eredményei eléggé gyengék. A PF esetén ez a mérték függ a ponthalmaz méretétől. Növelve a ponthalmaz méretét a labdavesztések száma növekszik, de ezt ellensúlyozza az, hogy az újramegtalálás sokkal gyorsabban megtörténik. Ezt a viselkedést azzal lehet magyarázni, hogy mivel az algoritmus több pontot generál, azok jobban szétszóródnak a tárgy központja körül és így nagyobb az esély arra, hogy közel kerüljenek a keresett tárgyhöz.

2. táblázat. Tárgy elvesztése

Alg.	zaj nélkül	zajjal
PF ₁₀	1.5	1.0
PF ₃₀	11.5	12.0
PF ₇₀	20.5	16.0
KF	5.5	9.5
UKF	3.0	5.0

A 2. táblázat arra vonatkozó statisztikákat tartalmaz, hogy hányszor veszíti el az algoritmus a labdát zajmentes, illetve zajos mozgás esetén. Az elvégzett kísérletek azt mutatják, hogy a zajos mozgás nem befolyásolja szignifikánsan a PF működését, sőt a PF₇₀ jobbnak bizonyult zajjal, mint anélkül. Ez nem mondható el a KF-ről és az UKF-ről, amelyeknél számottevő teljesítmény-visszaesés tapasztalható. Az UKF jobb volt, mint a KF, ez prognosztizálható volt a két megközelítés állapotterének a definíciójából.

3. táblázat. Tárgy elvesztése

Alg.	lineáris mozgás	nemlineáris mozgás
PF ₁₀	1.0	1.5
PF ₃₀	12.5	11.0
PF ₇₀	20.0	16.5
KF	2.0	13.0
UKF	1.5	6.5

A 3. táblázat azt tartalmazza, hogy hányszor veszítette el a labdát az algoritmus line-

áris és nemlineáris mozgások esetén. Az eredmények a 2. táblázatban található eredményekhez hasonlóak: a mozgás típusa nem lényeges a PF számára, míg számít a KF-UKF összehasonlításban. Számottevő visszaesés észlelhető a KF használatakor, míg az UKF jelentősen robusztusabb.

4. Következtetések

Mindenek előtt kijelenthető, hogy nem létezik univerzális tárgykövető algoritmus, nem várhatjuk el egyiktől sem, hogy ugyanolyan jól működjön minden körülmény között. Az algoritmus kiválasztásánál figyelembe kell venni a környezetet, amiben az illető algoritmust használni szeretnénk.

A Kálmán szűrő és az unscented Kálmán szűrő nyújtotta a legpontosabb becsléseket a tárgy pozícióját tekintve, ám bár gyakran csődöt mondott, ha nemlineáris mozgás vagy zaj lépett fel. Csak ezt a két módszert tekintve az unscented Kálmán szűrő által nyújtott megoldás minden tekintetben jobbnak bizonyult. A nemlinearitásból és a zajos mozgásból származó nehézségek megoldásaként a részecskeszűrő sikerrel alkalmazható. Gyengébb becslést eredményez, mint az előző megoldások, de a nemlinearitás és a zaj elhanyagolható mértékben befolyásolja a működését, továbbá skálázható a pontthalmaz méretének a változtatásával.

Nem meglepő, hogy a jobb eredményeket produkáló algoritmusok nagyobb számítási igényvel rendelkeznek.

A tárgyövetés témáját nem merítettem ki, érdekes lenne kipróbálni az említett algoritmusokat valós helyzetekben, amikor valódi roboton futnának és valós környezetben kellene tárgyakat követniük, úgy, hogy a feltételek nem lennének *ideálisak*, azaz a robot (kamera) mozgásában kontroll információk szólnának bele, illetve a zaj nem lenne független a mozgástól.

Hivatkozások

- A. Doucet, S. Godsill, és C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2005.
- D. Fox. Kld-sampling: Adaptive particle filters. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.
- N. Gordon, J. Salmond, és A. Smith. A novel approach to non-linear/non-gaussian bayesian state estimation. *IEEE Preceedings on Radar and Signap Processing*, pages 107–113, 1993.
- M. S. Grewal és A. P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. John Wilney and Sons, Inc., second edition, 2001.
- S. Haykin. *Kalman Filtering and Neural Networks*. John Wilney and Sons, Inc., 2001.
- M. Isard és A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. on Computer Vision*, volume 1, pages 343–356, Cambridge UK, 1996.
- M. Isard és A. Blake. Condensation – conditional density propagation for visual tracking. In *Int. J. Computer Vision*, volume 1, pages 5–28, 1998.
- R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- A. Kong, J. Lius, és W. Wong. Sequential imputations and bayesian missing data problems. *American Statistical Association* 89, pages 278–288, 1994.
- W. H. Press, B. P. Flannery, S. A. Teukolsky, és W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, second edition, 1992.
- M. I. Ribeiro. *Kalman and Extended Kalman Filters: Concept, Derivation and Properties*. Institute for Systems and Robotics, Instituto Superior Tecnico, February 2004.
- C. P. Robert és G. Casella. *Monte Carlo Methods*. Springer, second edition, 2004.

- J. J. Simon és J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. *The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II*, 1997.
- J. K. Uhlmann. *Dynamic map building and localization: New theoretical foundations*. PhD thesis, New theoretical foundations, 1995.
- R. van der Merwe, A. Doucet, N. de Freitas, és E. Wan. The unscented particle filter. In T. G. D. T. K. Leen és V. Tresp, editors, *Advances in Neural Information Processing Systems (NIPS13)*. MIT Press, December 2000.
- G. Welch és G. Bishop. An introduction to the kalman filter. In *TR*, 1995.