

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

**Netiesinių daugdarų atpažinimo metodų taikymo web-kamera gautiems
vaizdų rinkiniams analizuoti tyrimas**

Analysis of non-linear manifold learning methods applied on image collections
provided by webcam

Magistro baigiamasis darbas

Atliko:	Ignas Petrauskas	(parašas)
Darbo vadovas:	Prof. habil. dr. Gintautas Dzemyda	(parašas)
Recenzentas:	Doc. dr. Rimantas Vaicekauskas	(parašas)

Vilnius – 2012

Turinys

1.Įvadas.....	5
1.Darbo tikslai ir uždaviniai.....	6
1.1.Magistrinio darbo tikslas.....	6
1.2.Magistrinio darbo uždaviniai.....	6
2.Daugiamačių duomenų vizualizavimo metodai.....	6
2.1.Daugiamatės skalės metodas.....	7
2.2.Trianguliacija.....	8
2.3.Sammon'o algoritmas.....	8
2.4.Santykinės perspektyvos metodas.....	9
2.4.1.Daugdarų RPM metodas.....	10
2.5.Kreivinių komponentų analizė.....	11
2.6.Pagrindinių komponentų analizė.....	11
3.Netiesinių daugdarų atpažinimo metodai.....	12
3.1.Lokaliai tiesinis vaizdavimas.....	12
3.2.Laplaso matricos tikrinių žemėlapių metodas.....	13
3.3.Hesės matricos tikrinių žemėlapių metodas.....	14
3.4.Lokaliųjų liečiamųjų erdvių rikiavimas.....	14
3.5.SMACOF metodas.....	15
3.6.ISOMAP metodas.....	15
4.Tiriami duomenys.....	16
4.1.Elipsinis / apvalus judėjimas.....	23
4.1.1.SMACOF.....	23
4.1.2.Sammon.....	24
4.1.3.RPM.....	25
4.1.4.Manifold RPM.....	26
4.1.5.Isomap.....	27
4.1.6.CCA.....	28
4.1.7.PCA.....	29
4.2.Horizontalus ir vertikalus judėjimas.....	30
4.2.1.SMACOF.....	30
4.2.2.Sammon.....	31
4.2.3.RPM.....	32
4.2.4.Manifold RPM.....	33
4.2.5.Isomap.....	34
4.2.6.CCA.....	35
4.2.7.PCA.....	36
4.3.Judėjimas kvadratu.....	37

4.3.1.SMACOF.....	37
4.3.2.Sammon.....	38
4.3.3.RPM.....	39
4.3.4.Manifold RPM.....	40
4.3.5.Isomap.....	41
4.3.6.CCA.....	42
4.3.7.PCA.....	43
5.Praktinis Isomap algoritmą panaudojančios programos įgyvendinimas.....	44
5.1.Programos sandara.....	44
5.2.Programos algoritmas.....	45
5.3.Eksperimentai.....	46
6.Išvados ir rezultatai.....	53
7.Rekomendacijos.....	54
8.Literatūros sąrašas.....	55

Santrauka

Šiame darbe nagrinėjami netiesiniai daugdarų atpažinimo metodai ir daugiamačių duomenų projekcijos metodai. Siūloma jais spręsti keliais laisvės laipsniais besisukančio objekto orientacijos radimo problemą.

Aprašomi MDS, Trianguliacijos, Sammon, RPM, mRPM, CCA, PCA, LLE, LE, HLLE, LTSA, SMACOF ir Isomap metodai.

Su kai kuriais iš jų atliekami web-kamera gautų galvos atvaizdų tyrimai.

Isomap algoritmo pagrindu sukuriama programinė įranga ir su ja taipogi atliekami galvos orientacijos tyrimai.

Summary

This paper with Analysis of non-linear manifold learning methods and multidimensional data projection methods. It is proposed use them in solving problem of detection of orientation of object, moving in few degrees of freedom.

Methods described: MDS, triangulation, Sammon, RPM, mRPM, CCA, PCA, LLE, LE, HLLE, LTSA, SMACOF and Isomap.

Some of them are used to analyze head images acquired by webcam..

Application is created which is then used to analyze head orientation by implementing Isomap method.

1. Įvadas

Šiais laikais įvairiose mokslo srityse gaunamus didelius informacijos kiekius galime apdirbti ir ištirti tik kompiuteriais. Paprastai tie duomenys yra apimantys daug matavimų - daugiamatai - juos apibūdina labai daug skirtingų parametrų.

Pavyzdžiui, duomenų rinkiniai gali būti tokie:

- daug skirtingų objektų, tarp kurių reikia rasti skirtumus ar panašumus ir pagal tai juos paskirstyti į atitinkamas grupes, arba išskirti ypatingai neatitinkančius vidurkio. Tai gali būti skirtingų institucijų veiklos ir rezultatų duomenys, vietovių geologiniai duomenys ir t.t.;
- didelis kiekis to pačio objekto atvaizdų, tik pasukto skirtingais kampais, ir reikalinga iš jų išgauti kokią nors naudingą informaciją.

Kaip žinome, kiekvienas skaitmeninis atvaizdas yra sudarytas iš duomenų masyvo, tai yra - kiekvieną pikselį apibūdina jo koordinatės ir jame esanti informacija - spalvų kiekiai (raudonos, žalios ir mėlynos, arba žydros, purpurinės, geltonos ir juodos; spalvoto atvaizdo atveju) arba pikselio gylis (nuo 0 iki 255, nespalvoto 8 bitų spalvų gylio vaizdo atveju).

Taigi, kiekvieną vaizdą galime laikyti daugiamatium objektu, kuris turi tiek matavimų, kiek ir pikselių. Kitaip, nei kitus daugiamatius duomenis, skirtumus ir panašumus tarp vaizdų žmogaus akis gali lengvai atpažinti, bet iš jų gauti reikšmingą informaciją skaitmeniniam įrenginiui yra nepalyginamai sudėtingiau.

Vienas būdų daugiamatiams duomenims pažinti, yra jų vizualizavimas.

1. Darbo tikslai ir uždaviniai

1.1. Magistrinio darbo tikslas

Ištirti galimybę įvertinti su keliais laisvės laipsniais besisukančio objekto orientaciją web kamera fiksuojamų vaizdų pagrindu naudojantis netiesinių daugdarų atpažinimo metodais.

1.2. Magistrinio darbo uždaviniai

- Apžvelgti daugiamačių duomenų vizualizavimo metodus.
- Apžvelgti netiesinių daugdarų atpažinimo metodus.
- Susigeneruoti testinius judančio objekto duomenis, kuomet objekto fonas yra vienalytis arba užpildytas kitais nejudančiais objektais.
- Pasirinkti tiriamus duomenis.
- Pasirinkti tyrimams daugiamačių duomenų vizualizavimo metodus.
- Eksperimentinio tyrimo atlikimas.
- Programos, naudojančios tinkamiausią metodą, kūrimas.
- Eksperimentai su sukurta programa.
- Išvadų iš eksperimentinio tyrimo apibendrinimas.
- Rekomendacijų rengimas.

2. Daugiamačių duomenų vizualizavimo metodai

Projekcijos metodai yra daugiamačių duomenų vizualizacijos metodai, kurie gali sumažinti duomenų dimensiškumą iki žmogui suprantamo lygio - dviejų ar trijų dimensių atvaizdo. Vizualizavimas - duomenų pateikimas grafiškai – tai įsisavinti žmogui yra lengviau, nei tekstinius duomenis dėl smegenų savybių, kurių čia neaptarinėsime.

Mažinant duomenų dimensiškumą kyla tokios pagrindinės problemos:

- iškraipymai dėl dimensijos sumažinimo;

- reikalinga, kad duomenų projekcijos, atvaizduojant iš didesnės į mažesnę dimensiją, nepersidengtų, taip apsunkinant informacijos atpažinimą;
- reikalinga išlaikyti objektų panašumus ir skirtumus.

Egzistuoja tokios projekcijos metodų grupės:

- tiesinės projekcijos metodai: pagrindinių komponentų analizė (PCA) ir t.t. Šiais metodais ieškoma tiesinės tiriamų duomenų transformacijos;
- netiesinės projekcijos metodai: lokaliai tiesinis vaizdavimas (LLE), Sammon'o metodas (Sammon mapping) ir t.t. Šiais metodais ieškomos netiesinės tiriamų duomenų transformacijos. Šie metodai yra sudėtingesni, nei tiesinės projekcijos metodai, nes reikalauja daugiau skaičiavimų, bet jie yra tikslesni, nes paprastai išlaiko tikslesnius atstumus tarp taškų.

Pažymėkime daugiamačių taškų aibę: $X = \{X_1, \dots, X_m\}$, $X_i = (x_{1i}, \dots, x_{ni})$.

Aibės X projekcija d -matėje erdvėje (mažesnio matavimo skaičiaus erdvėje):

$$Y = \{Y_1, \dots, Y_m\}, Y_i = (y_{1i}, \dots, y_{di}).$$

Taigi, vizualizavimo tikslas yra, žinant aibę X , gauti aibę Y .

2.1. Daugiamatės skalės metodas

Daugiamatės skalės metodas (MDS – Multi Dimensional Scaling), [Sac01] mažindamas duomenų dimensiškumą ir ieškodamas objektų projekcijų žemesnės dimensijos erdvėje, stengiasi išdėstyti panašesnius objektus arčiau, ne tokius panašius – toliau vienus nuo kitų, naudodamasis nepanašumų matrica:

$$\Delta := \begin{pmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,I} \\ d_{2,1} & d_{2,2} & \dots & d_{2,I} \\ \vdots & \vdots & & \vdots \\ d_{I,1} & d_{I,2} & \dots & d_{I,I} \end{pmatrix}.$$

Čia d_{ij} yra, pavyzdžiui, atstumas tarp taškų X_i ir X_j , $X_i = (x_{1i}, \dots, x_{ni})$.

Algoritmo tikslas yra rasti taškus $Y_i = (y_{1i}, \dots, y_{di})$, kurie atitiktų minimumą kokios

$$\min_{x_1, \dots, x_I} \sum_{i < j} (d_{i,j}^* - d_{i,j})^2.$$

nors paklaidos funkcijos, pvz.:

$$, \text{ čia } d^* = \|Y_i - Y_j\|$$

2.2. Trianguliacija

Trianguliacijos metodu [LSB77], daugiamatės erdvės taškai atvaizduojami į dvimatę erdvę. Šio metodo ypatybė, kad taškai visad atvaizduojami vienodais atstumais nuo dviejų prieš tai atvaizduotųjų taškų.

Kurie tai taškai, gali būti nustatoma dviem metodais:

- Sekančio artimiausio kaimyno (Second nearest neighbor) – tiksliausiai išlaikomi atstumai nuo taško iki dviejų artimiausių jo kaimynų.
- Atramos taško (Reference point) – tiksliausiai išlaikomi atstumai nuo taško iki pirmo arčiausio kaimyno ir anksčiau pasirinkto bendro atramos taško.

Antrasis metodas padeda atvaizduoti įvairius duomenis iš tam tikro taško perspektyvos, kas gali padėti juos paprasčiau suprasti.

2.3. Sammon'o algoritmas

Sammon'o algoritmas [Sam69] netiesiškai atvaizduoja daugiadimensinę erdvę į mažiau dimensijų turinčią. Jo ypatybė - stengiamasi išlaikyti atstumus, esančius tarp taškų daugiau dimensijų turinčioje erdvėje, proporcingus, projektuojant juos į mažiau dimensijų turinčią erdvę.

Sammon'o projekcija turi minimizuoti paklaidą, vadinama Sammono apkrova:

$$E = \frac{1}{\sum_{i < j} [d_{ij}^*]} \sum_{i < j}^N \frac{[d_{ij}^* - d_{ij}]^2}{d_{ij}^*}$$

Čia d_{ij} – atstumas tarp taškų didesnės dimensijos erdvėje, d_{ij}^* - mažesnės dimensijos erdvėje.

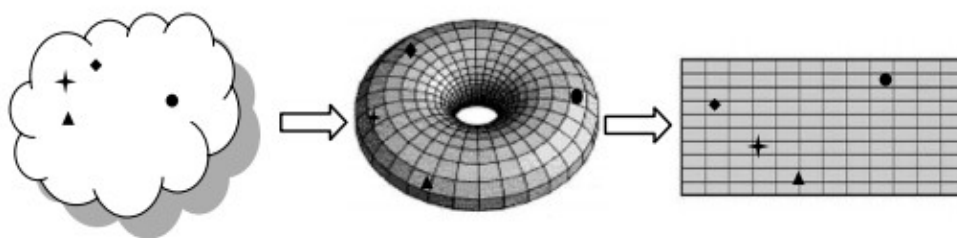
Paklaida minimizuojama gradientinio nusileidimo metodu arba kokiu nors iteraciniu metodu.

2.4. Santykinės perspektyvos metodas

Santykinės perspektyvos metodas (RPM - Relational Perspective Map) yra skirtas atvaizduoti taškus, esančius daugiadimensinėse erdvėse. [Li04]

Pirmiausia algoritmas atvaizduoja taškus ant toro paviršiaus, tada išlanksto toro paviršių į stačiakampį, perpjaunant jį vertikaliai ir horizontaliai.

Ši procedūra pavaizduota pirmame paveiksle.



1 pav. RPM procedūra (šaltinis:

<http://iihm.imag.fr/daassi/papierInfoVisJournal/papierExemple/9500051a.pdf>)

Kad rasti geriausią atvaizdavimą, RPM algoritmas apibrėžia energijos funkciją:

$$E_p := \sum_{i < j} \frac{\delta_{ij}}{pd_{ij}^p}$$

kur p yra algoritmo parametras, vadinamas kietumu, δ_{ij} yra geodezinis atstumas tarp dviejų atvaizdo taškų toro paviršiuje ir d_{ij} yra atstumas tarp taškų X_i ir X_j aukštesnės dimensijos erdvėje. Tada RPM algoritme naudojamas laipsniško nusileidimo metodas, kad rasti žemiausios energijos išsidėstymą.

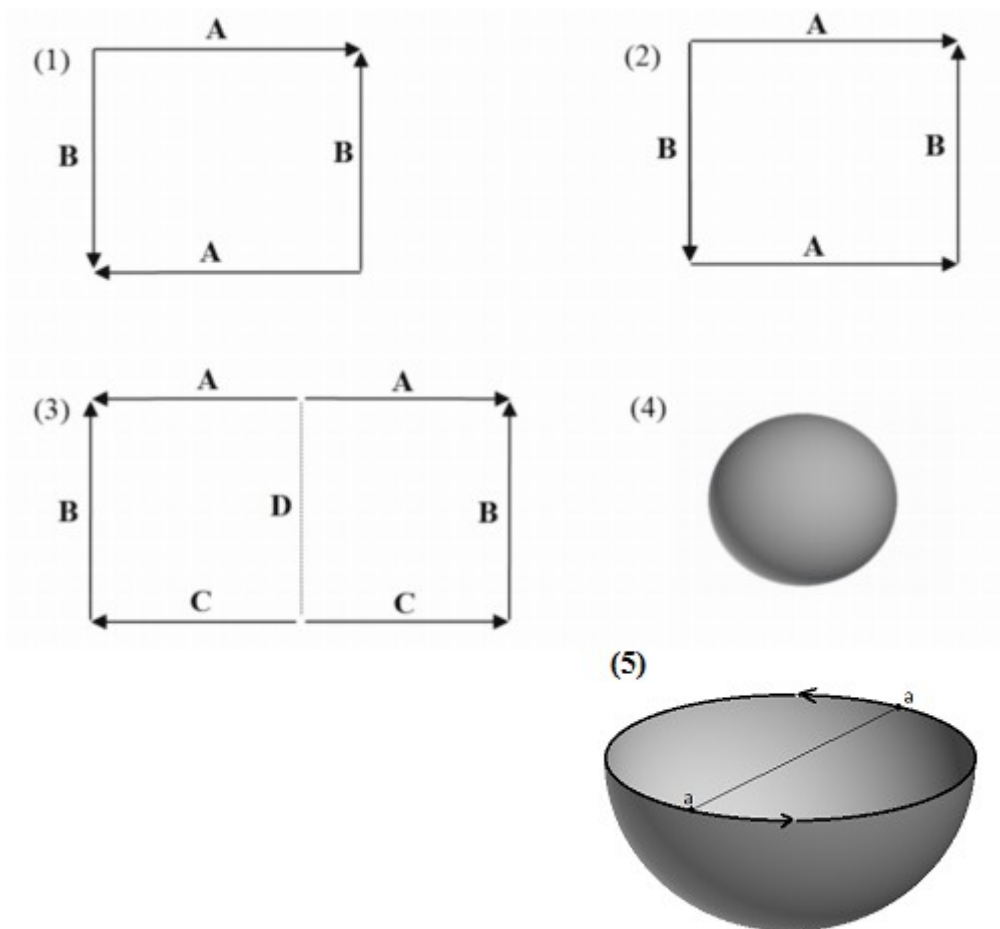
Optimizuojant reikalinga sumažinti energiją E_p simuliuojant dinamišką sistemą, varomą aukščiau aprašytos energijos. Kadangi taškai, esantys santykinai toliau, atitinka didesnes atstumiančiąsias jėgas ant toro paviršiaus, jie turi būti atvaizduojami toliau vienas nuo kito.

Reikia įsivaizduoti, kad gautas išsidėstymas nėra stačiakampis, o tiesiog jo kraštai yra sujungti, kaip pradinėje (toro) figūroje.

2.4.1. Daugdarų RPM metodas

Egzistuoja dar vienas reikšmingas RPM metodo variantas – daugdarų santykinės perspektyvos metodas (Manifold RPM), kuris vietoje toro paviršiaus naudoja kitų objektų paviršius (2 pav.) :

1. Plokščia reali projekcinė plokštuma – keturkampis, kurio priešingos kraštinės susijungia, vieną iš jų pavertus 180 laipsnių kampu.
2. Plokščias Klein'o butelis – keturkampis, kurio dvi priešingos kraštinės susijungia, kitos dvi – susijungia, vieną iš jų pavertus 180 laipsnių kampu.
3. Plokščia sfera – keturkampis, kurio abiejų pusių kraštinės susisiečia.
4. Trimatė sfera – rutulio paviršius.
5. Projekcinis pusrutulius – pusrutulius, kurio priešingi kraštiniai taškai susijungia, ši daugdara topologiškai identiška pirmajai.



2 pav. Manifold RPM atvaizdavimo objektai

2.5. Kreivinių komponentių analizė

Kreivinių komponentių analizė (CCA - Curvilinear Component Analysis) yra patobulintas Sammon'o metodas [LLDV00]. Jis teikia pirmenybę lokaliai topologijai, pirmiausia atkurdamas trumpesnius atstumus, po to - didesnius.

Taigi, paklaidos formulė pasidaro tokia:

$$E_{CCA} = \sum_{i,j=1}^N (d_{i,j} - d_{i,j}^*)^2 F_{\lambda}(d_{i,j}^*)$$

Čia d_{ij} – atstumas tarp taškų didesnės dimensijos erdvėje, d_{ij}^* - mažesnės dimensijos erdvėje. Lambda yra nuo laiko priklausomas parametras, kuris palaipsniui keičiasi nuo pradžioje duotojo lambda t iki 0, optimizacijos proceso metu.

2.6. Pagrindinių komponentių analizė

Pagrindinių komponentių analizė (PCA - Principal Component Analysis) siekia sumažinti duomenų dimensiškumą ir kuo didesnę dispersiją, tiesiškai juos transformuojant ir atmetant tuos gautuosius komponentus, kurių maža dispersija. [Jo89]

Palyginus su kitais metodais, PCA dimensijų kiekį sumažina greičiau, bet gali atlikti tik tiesines transformacijas, taigi yra ne toks efektyvus atskiriant netiesines ypatybes, kaip kiti, netiesiniai, algoritmai.

Per centrinį (mažiausią atstumą nuo visų duomenų taškų erdvėje turintį) tašką išvedama tiesė per didžiausią dispersiją turinčią duomenų dalį. Iš to pačio taško išvedama kita tiesė, statmena pirmajai. Tos tiesės vadinamos pagrindinėmis komponentėmis.

Algoritmo vykdymo metu iš duomenų sudaroma kovariacinė matrica, iš kurios apskaičiuojami tikrinės reikšmės ir vektoriai. Iš surūšiuotų tikrinių vektorių gaunama pagrindinių komponentių matrica.

Pagrindinių komponentių matrica paprastai sudaroma iš kažkokio kiekio d tikrinių vektorių, dažniausiai turinčių didžiausias tikrines reikšmes, nes kiti tikriniai vektoriai neša per mažai informacijos (turi per mažą įtaką rezultatui).

3. Netiesinių daugdarų atpažinimo metodai

Daugdara - tai aibė, kurios kiekvieno taško aplinkoje yra įvesta koordinačių sistema. Nors daugdara apibendrina bet kokios dimensijos erdvėje kreivės ir paviršiaus sąvokas, bet lokaliai daugdara nesiskiria nuo euklidinės erdvės, nes yra padaryta iš euklidinės erdvės gabaliukų. [AT07]

Netiesines daugdaras galima atpažinti todėl, kad dažnai daugiadimensiniuose duomenyse „slepiasi“ mažesnio dimensiško daugdara, netiesinių daugdarų atpažinimo metodų tikslas ir yra tas daugdaras atrasti.

Šie metodai išlaiko daugdaros topologiją, tai yra - kiekvieno taško kaimynai didesnės dimensijos erdvėje lieka tokie patys ir mažesnės dimensijos erdvėje. Kartais atstumai išlaikomi ne tik lokaliai, bet ir globaliai (pavyzdžiui, ISOMAP metodas išlaiko globalią struktūrą).

Vaizdų atpažinimo atveju, kai tiriami daugiamaciai duomenys yra paveikslėlių rinkinys, didžioji dimensijų dalis mums nieko nepasako ir mums yra reikalinga nustatyti žymiai mažesnę svarbių dimensijų kiekį - vadinamuosius laisvės laipsnius - kaip objektas gali sukinėtis erdvėje.

3.1. Lokaliai tiesinis vaizdavimas

Lokaliai tiesinis vaizdavimas (LLE – Local Linear Embedding) [RS00] iš tikro priklauso netiesinių metodų grupei. Šiuo metodu mažinant duomenų daugiadimensiškumą, išlaikomi tik artimiausi vaizduojamo taško kaimynai. Metodas išlaiko globalią struktūrą.

- Pirmiausia randami visi taško kaimynai pagal euklidinį atstumą, ieškant k artimiausių kaimynų (svarbu rasti tinkamiausią kiekvienam atvejui k) arba juos paimant iš tam tikro spindulio rutulio aplink tašką.
- Tada visi erdvės taškai išreiškiami savo kaimynų kombinacijos formule radus optimalias svorių reikšmes.
- Galiausiai su tais svoriais apskaičiuojama projekcija.

Metodo privalumai:

- MDS metodais stengiamasi išlaikyti atstumus tarp visų taškų. LLE nereikalauja išlaikyti atstumų tarp labiausiai nutolusių taškų.
- LLE metodu mažinant dimensiško daugdara, galima identifikuoti pirmiau nematomą daugdaros struktūrą.

Šis metodas dažnai naudojamas veidams atpažinti. [MBT05] [ZZ06] [ZZL05] [HKP02]

3.2. Laplaso matricos tikrinių žemėlapių metodas

Laplaso matricos tikrinių žemėlapių (LE – Laplacian Eigenmap) metodas [BN01] taip pat mažina duomenų daugiadimensiškumą, išlaikydamas daugdaros lokalias struktūras.

Šis metodas yra Lokaliai tiesinio vaizdavimo metodo variantas.

Metodas sukuria grafą, kurio kiekviena viršūnė yra duomenų taškas ir kurie yra sujungti pagal kaimyninių taškų artumą – kainos funkcijos minimizacija užtikrina, kad taškai, esantys netoli vienas kito daugdaroje, atvaizduojami arčiau vienas kito žemesnės dimensijos erdvėje, išsaugant lokalius atstumus.

Šio metodo algoritmas toks:

- Iš vizualizuojamos aibės X visų taškų sukonstruojamas kaimynystės grafas su m viršūnių ($m = X$ aibės taškų kiekis) ir kaimynus jungiančiomis briaunomis.

Kaimyniškumas nustatomas:

- ϵ -kaimynai: viršūnės i ir j yra sujungtos briauna, jei $\|X_i - X_j\|^2 < \epsilon$.
- n artimiausi kaimynai: viršūnės i ir j sujungtos briauna, jei i yra tarp n artimiausių j kaimynų ir atvirkščiai.
- Parenkami briaunų svoriai:
 - jei i ir j sujungtos briauna,

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}}$$

jei nesujungtos, svoris W_{ij} lygus 0.

- arba jei i ir j sujungtos briauna, svoris W_{ij} lygus 1, kitu atveju – 0.
- Randamos daugiamatės erdvės taškų projekcijos d -matėje erdvėje iš šio grafo Laplaso matricos (matricos, išreiškiančios grafą) tikrinių vektorių (eigenvectors).

3.3. Hesės matricos tikrinių žemėlapių metodas

Hesės matricos tikrinių žemėlapių (HLLE – Hessian Locally Linear Embedding) metodas remiasi Laplaso matricos tikrinių žemėlapių metodu, tik vietoje Laplaso matricos naudojama Hesės matrica. Ši matrica aprašo daugdaros kreivumą prie erdvės taškų. (matrica, kuri aprašo lokalias daugelio kintamųjų funkcijos kreives)

Šis metodas reikalauja ypač daug skaičiavimo resursų, dėl to netinka dideliems duomenų kiekiams apdoroti.

Algoritmas:

1. Randami kiekvieno taško k artimiausių kaimynų pagal euklidinį atstumą.
2. Kiekviename taške apibrėžiama lokalias liečiamosios erdvės bazė ir randamos konkretaus taško kaimynų projekcijos toje liečiamojoje erdvėje.
3. Pagal gautas kaimyninių taškų koordinates, apskaičiuojama matrica H_i , kuri aproksimuoja Hesės matricą tame taške.
4. Iš matricų H_i sudaroma matrica, kuri aproksimuoja Hesės matricą visai daugdarai.
5. Randamos daugiamatės erdvės taškų projekcijos mažesnės dimensijos erdvėje, atliekant H tikrinių vektorių ir juos atitinkančių tikrinių reikšmių (eigenvalues) analizę. Ieškoma d (mažesnės dimensijos kiekis) mažiausias, nelygias nuliui tikrines reikšmes atitinkantys vektoriai, kurie suformuoja mažesnės dimensijos taškų projekcijų koordinates.

3.4. Lokaliųjų liečiamųjų erdvių rikiavimas

Lokaliųjų liečiamųjų erdvių rikiavimas (LTSA – Local Tangent Space Alignment) yra metodas, kuris netiesiniu būdu atvaizduoja daugiadimensinius duomenis į žemesnę dimensiją, kaip ir kiti metodai. Kaip ir HLLE metodas, jis apibrėžia kiekvieno tiriamos erdvės taško lokalias savybes (lokaliąją liečiamąją erdvę). Šio metodo atveju, tikimasi, kad, kai daugdara teisingai išlankstoma, tai lokalias liečiamosios erdvės susirikiuos. Kiekvienam taškui randami k kaimynų ir toje kaimynystėje randamos lokalias erdvės. Kaip ir kituose metoduose, atvaizdavimo kokybė priklauso nuo tinkamai parinkto kaimynų kiekio k .

3.5. SMACOF metodas

Šis (SMACOF – Scaling by Majorizing a Complicated Function) metodas [BG97] plačiai naudojamas, minimizuoti daugiamatės skalės paklaidos (stress) funkciją:

$$\sigma(X) = \sum_{i < j \leq n} w_{ij} (d_{ij}(X) - \delta_{ij})^2 = \sum_{i < j} w_{ij} \delta_{ij}^2 + \sum_{i < j} w_{ij} d_{ij}^2(X) - 2 \sum_{i < j} w_{ij} \delta_{ij} d_{ij}(X)$$

Čia d_{ij} ir δ_{ij} yra atitinkamai santykinis atstumas ir atvaizdavimo atstumas tarp dviejų kūnų i ir j .

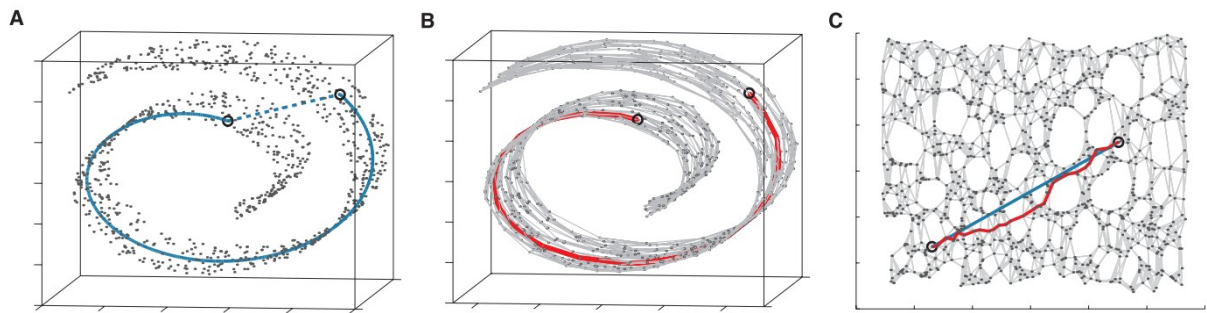
Atvaizdai, išgauti šio algoritmo, leidžia pamatyti duomenų rinkinių globaliai užimamas erdvės dalis. Beveik tą patį daro ir PCA ir Sammon'o metodai, tik kad PCA metodas gali būti taikomas tik duomenų rinkiniams su euklidiniais atstumais.

Vienintelis reikšmingas SMACOF metodo parametras tėra sustabdymo parametras, nuo kurio priklauso gautų žemėlapių kokybė. Kaip ir PCA algoritmas, SMACOF daugiau koncentruojasi į globalų vaizdą, kai Sammon'o metodas – daugiau į lokalių.

3.6. ISOMAP metodas

Izometrinis požymių vaizdavimas [Yan02] yra plačiai taikomas daugdarų atpažinimo metodas, kuris išlaiko ir lokalią ir globalią daugdaros struktūrą. Šis metodas naudoja geodezinius atstumus, nes euklidinis atstumas netiesinėse daugdarose negali tiksliai apibrėžti atstumo tarp taškų. Jis paremtas ir išplečia daugiamatės skalės metodus, kuriuose naudojamas euklidinis atstumas.

- Sukonstruojamas kaimynų žemėlapis - grafas (pagal k artimiausių kaimynų arba tam tikro skersmens rutulyje aplink tašką esančių kaimynų metodą, abiem atvejais šis žingsnis atliekamas naudojant euklidinius atstumus tarp taškų).
- Tarp visų taškų porų apskaičiuojami geodeziniai atstumai, naudojant Dijkstra arba Floyd algoritmus.
- Daugiamačių skalių metodu randamos projekcijos į mažesnę dimensiją.



3 pav. Euklidiniai ir geodeziniai atstumai.

Trečiame paveiksle galime matyti:

- A. kaip euklidinis (mėlynas brūkšniuotas) atstumas, skiriasi nuo geodezinio (mėlynas vientisas) atstumo;
- B. kaip geodezinis atstumas randamas grafe;
- C. tikrojo geodezinio ir metodo rasto geodezinio atstumo atvaizdavimą ISOMAP metodo veikimo pabaigoje.

4. Tiriama duomenys

Tiriama duomenys yra žmogaus galvos, nufotografuotos web-kamera ir pasuktos įvairiomis kryptimis, atvaizdai.

Duomenų rinkiniai yra tokie:

1. Galvos nuotraukos, už nugaros esant fonui, užpildytam kitais, nejudančiais, objektais.
2. Galvos nuotraukos, už nugaros esant vienalyčiam fonui.

Duomenų rinkiniai yra atskiros duomenų bazės, kuriose įrašytos nuotraukų sekos, kiekvieną nuotrauką išreiškus kableliais atskirtomis reikšmėmis (Comma separated values – CSV) – kiekvienas pikselis turi atskirą reikšmę nuo 0 iki 255 – pilkumo lygis, kadangi nuotraukos yra nespaltotos.

Kadangi vykdant eksperimentus algoritmams kartais pritrūkdavo atminties, teko sumažinti atvaizdų rezoliuciją nuo 640*480 taškų iki 320*240 taškų, nes reikšmingų skirtumų tarp rezultatų nebuvo pastebėta.

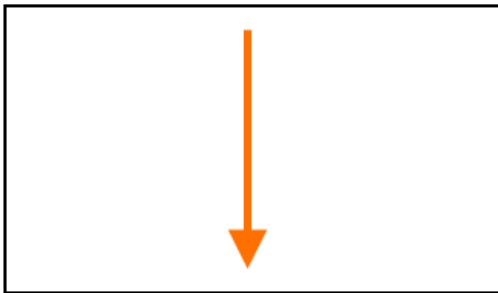
Eksperimentai atlikti su atskiromis duomenų bazėmis kiekvienam duomenų rinkiniui:

1. Galvos judesys, stebint ekraną vertikalčiai.

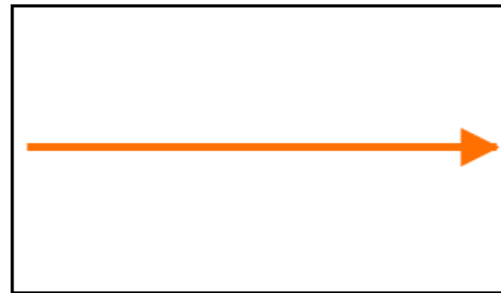
2. Galvos judesys stebint ekraną horizontaliai.
3. Galvos judesys stebint ekraną palei jo kraštus.
4. Galvos judesys stebint ekrane esančią elipsę.

Vizualinis paaiškinimas pateiktas 4 paveiksluke.

1.



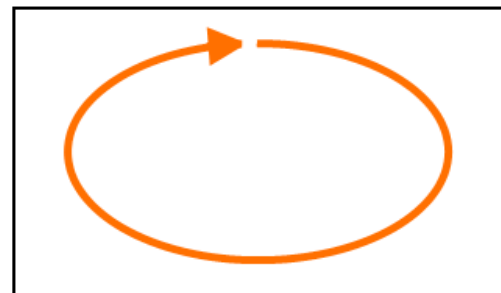
2.



3.

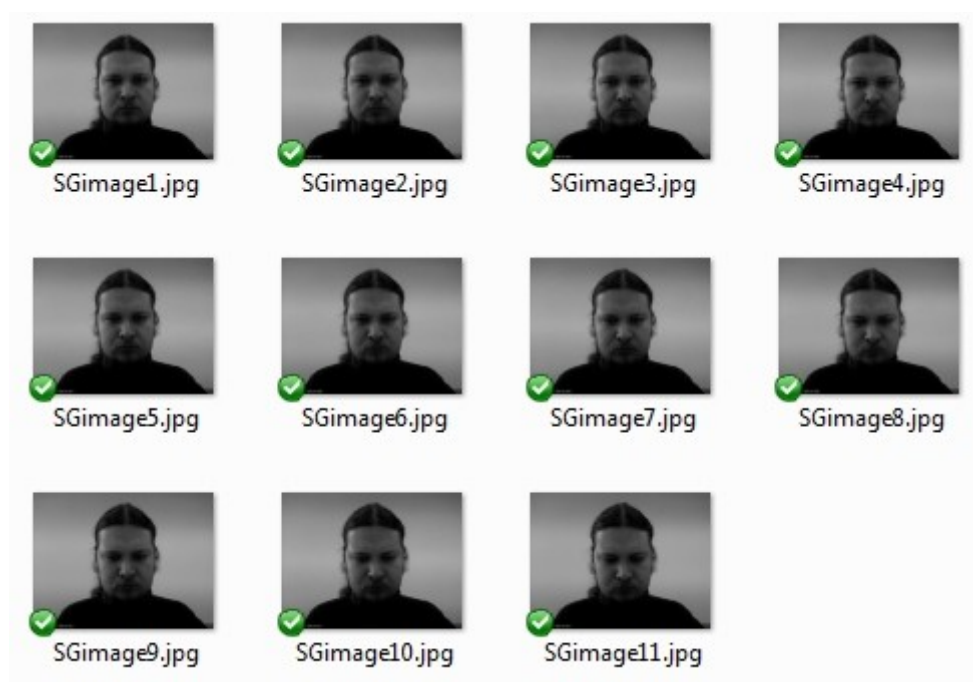


4.

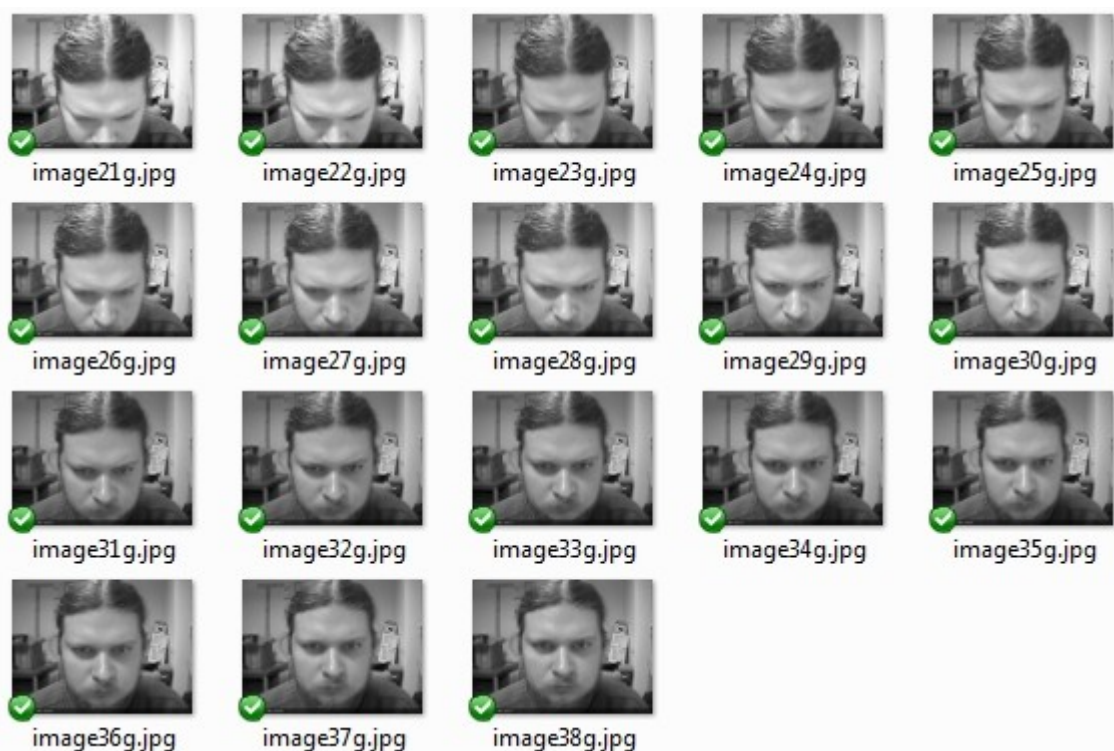


4. Pav.: Ekranu stebėjimo kryptys

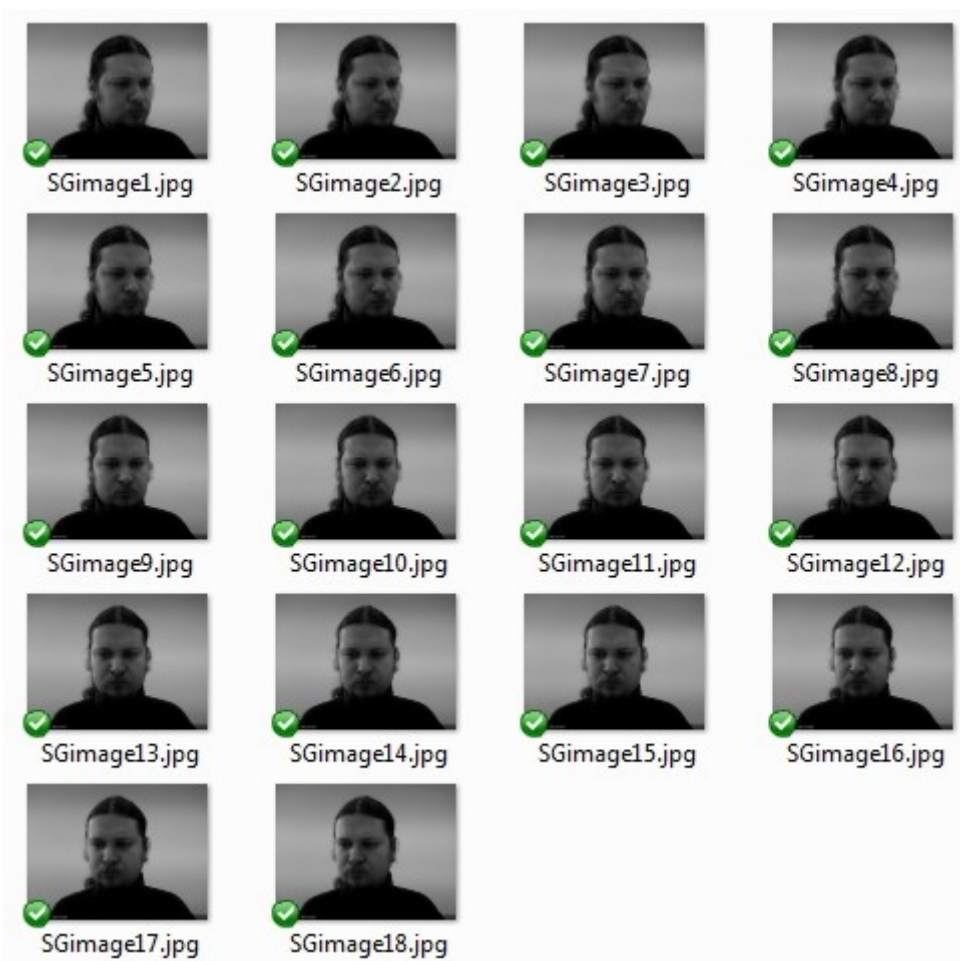
Toliau (3.2-3.9) pavaizduoti atvaizdai iš atvaizdų duomenų bazių.



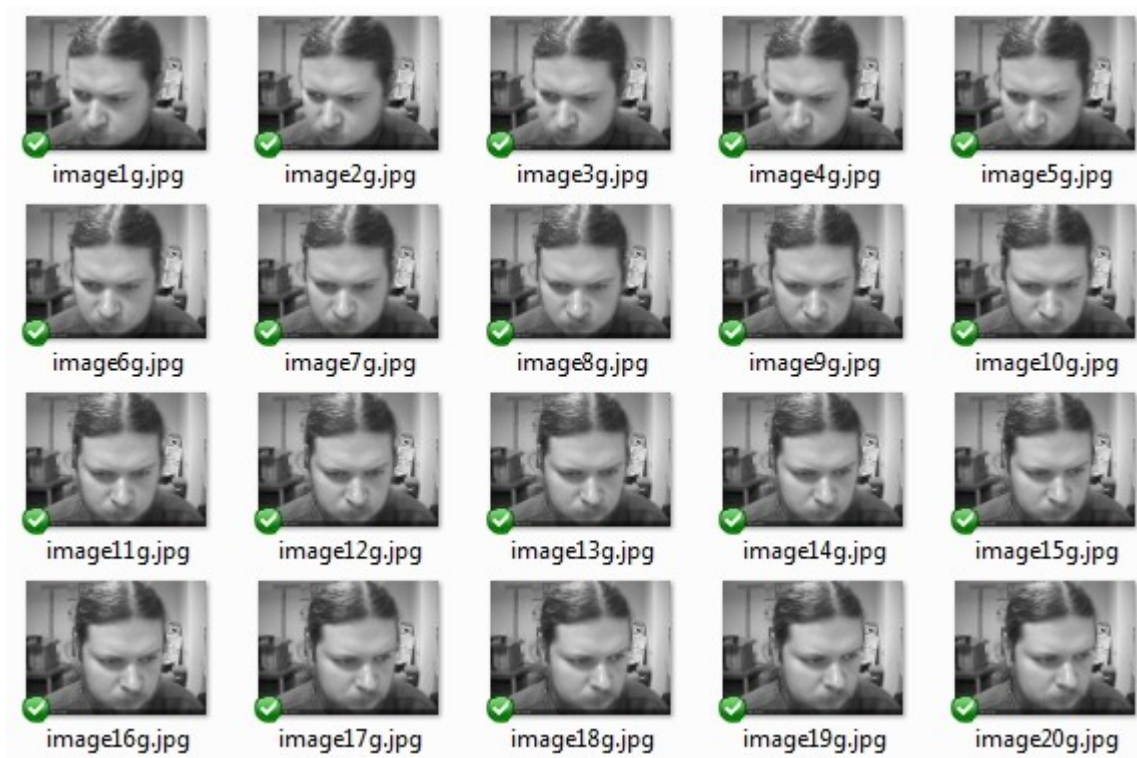
5. Pav.: Ekranu stebėjimas vertikaliai, fonas vientisas



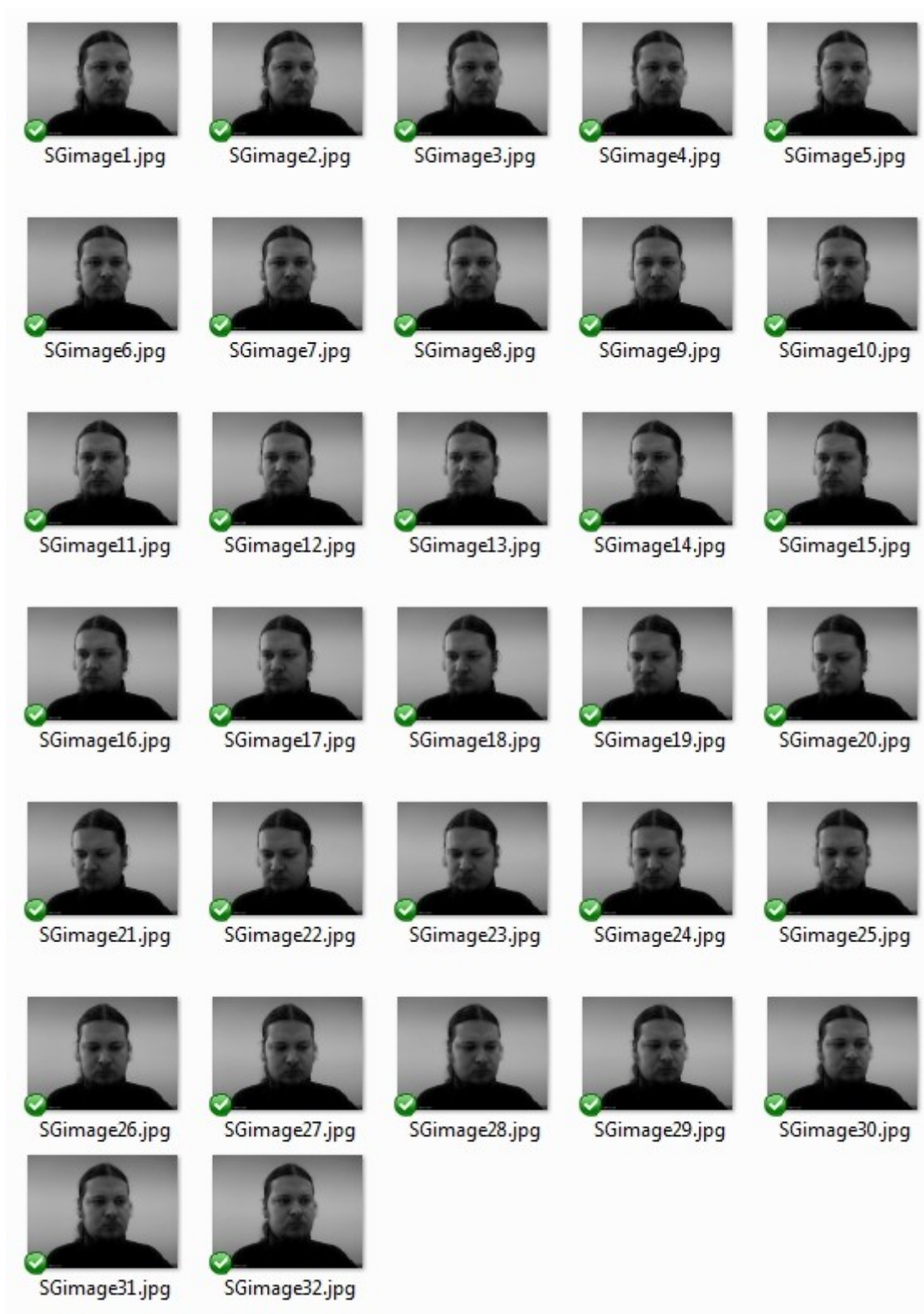
6. Pav.: Ekranu stebėjimas vertikaliai, fonas ne vientisas



7. Pav.: Ekrano stebėjimas horizontaliai, fonas vientisas



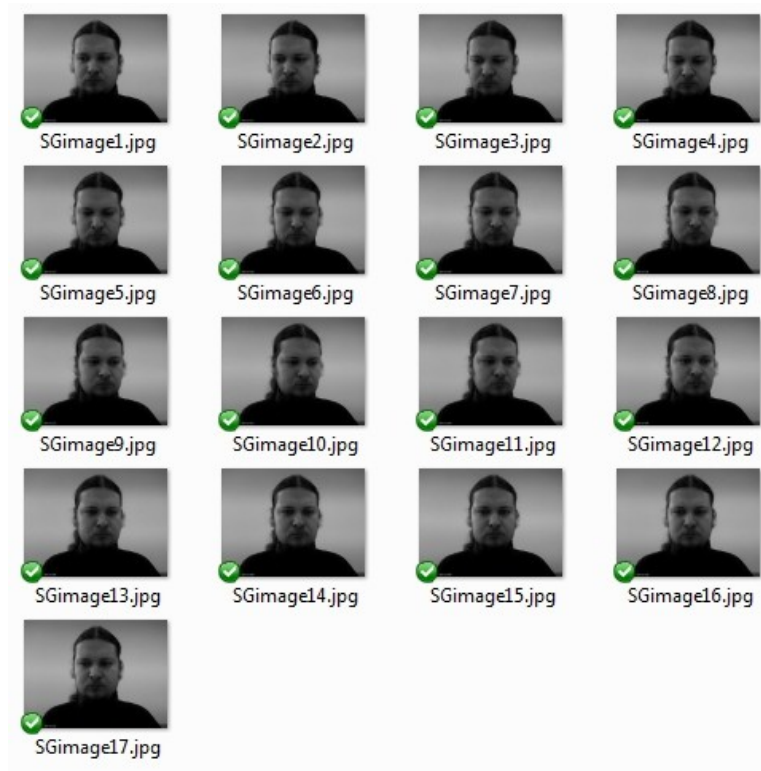
8. Pav.: Ekrano stebėjimas horizontaliai, fonas nevientisas



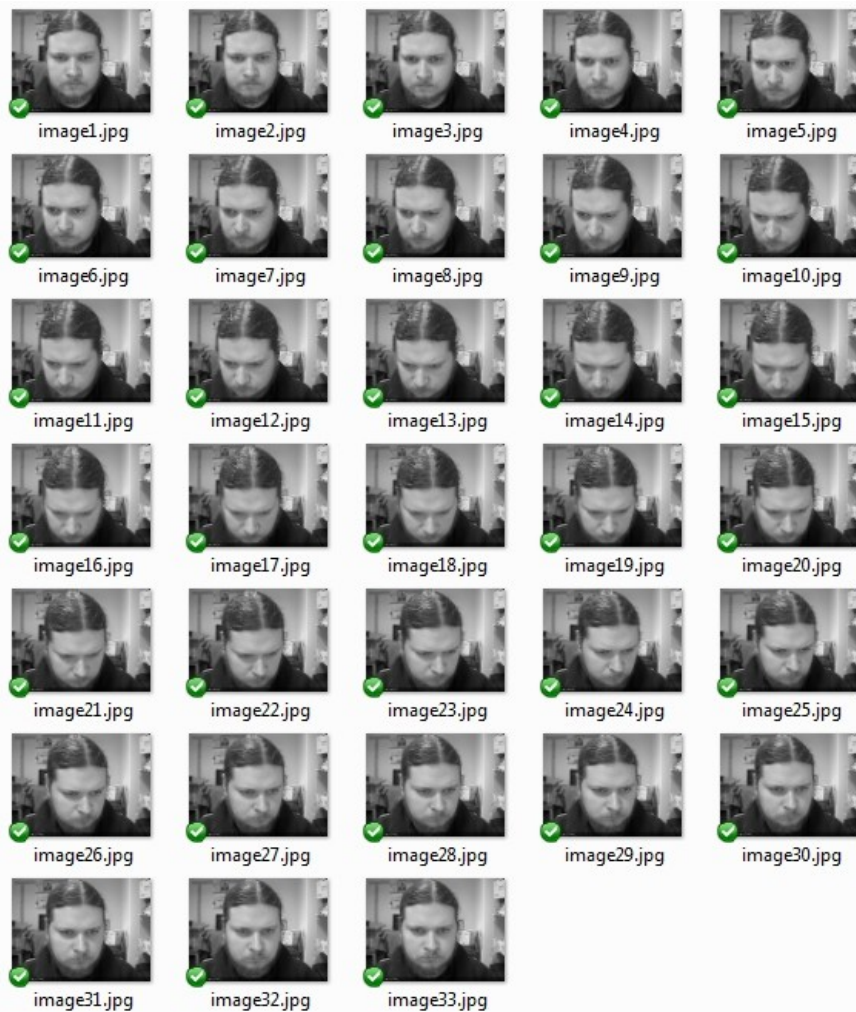
9. Pav.: Ekrano stebējimas kvadratu, fonas vientisas



10. Pav.: Ekranu stebējamas kvadratu, fonas nevientisas



11. Pav.: Ekrano stebējimas elipse, fonas vientiskas



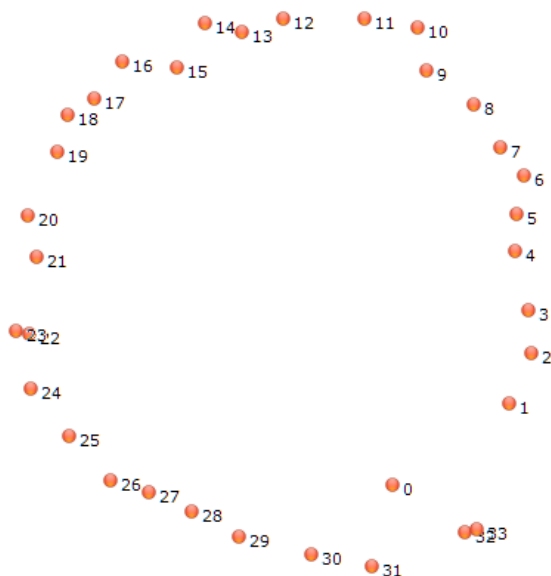
12. Pav.: Ekrano stebējimas elipse, fonas nevientiskas

4.1. Elipsinis / apvalus judėjimas

4.1.1. SMACOF



13. Pav.: SMACOF, vientisas fonas

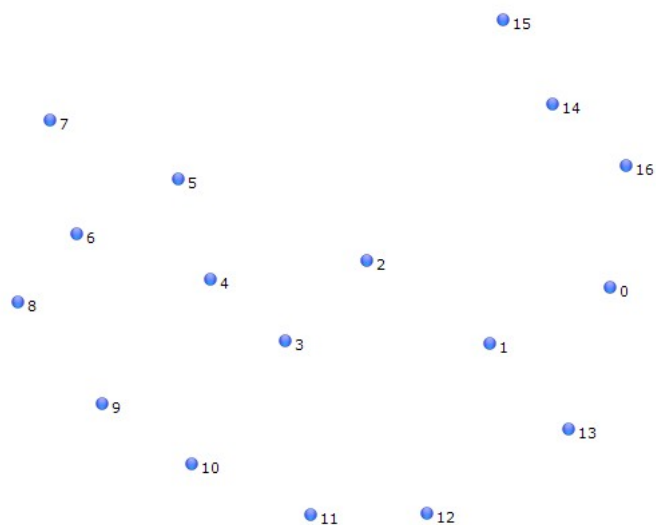


14. Pav.: SMACOF, nevientisas fonas

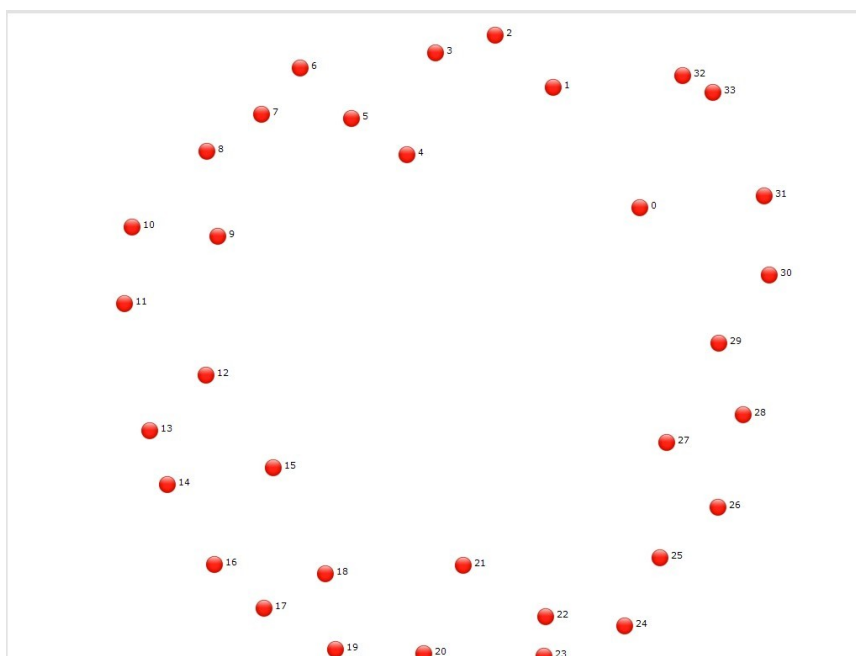
Šio algoritmo rezultatai, atpažįstant judesį apie elipsę, geri.

Ir pirmu (13 pav.), ir antru (14 pav.) atveju matome rezultatų diskretiškumo įtaką duomenų projekcijų atvaizdavimui, rečiau darant nuotraukas (kaip pirmu atveju), tarpai taip projekcijų atvaizduojami didesni. Judesys atvaizduotas pakankamai tolygiai, galime atpažinti jį vizualiai.

4.1.2. Sammon



15. Pav.: Sammon, vientisas fonas



16. Pav.: Sammon, nevientisas fonas

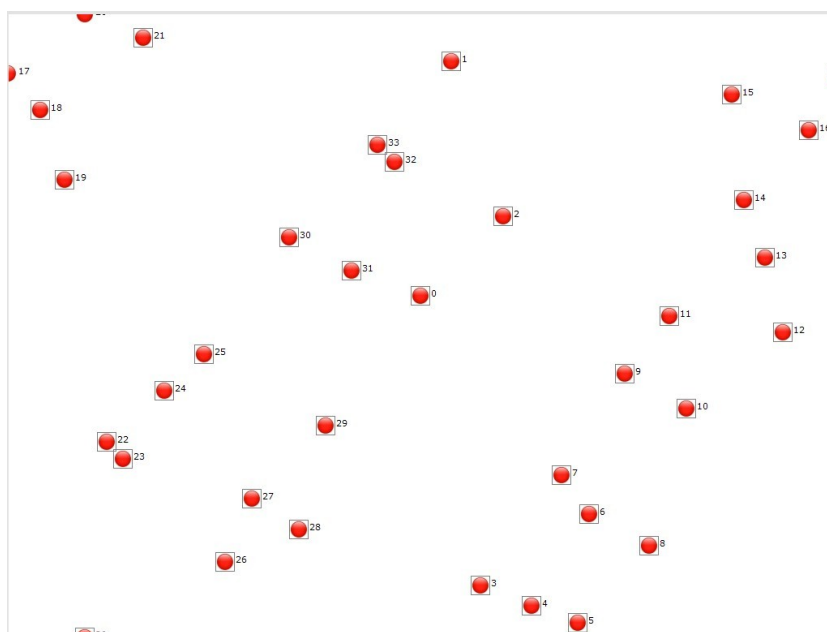
Pirmuoju (15) atveju Sammon'o algoritmas nesusiejo judesio pradžios ir pabaigos. Kadangi antrajame (16) atvejuje rezultatas palyginti aiškesnis, galima spėti, kad galvai užimant mažiau vietos atvaizde, nepakako duomenų susieti pirmo ir paskutinio atvaizdų. Galbūt būtų įmanomą rezultatą pagerinti, parenkant didesnio diskretiškumo duomenys. Atvaizdus atpažįstant gyvai, galbūt ši problema išsispręstų.

Antruoju atveju rezultatas vizualiai labiau atitinka norimą - duomenų kiekis pakankamas, taipogi galva užima didesnę atvaizdo dalį.

4.1.3. RPM



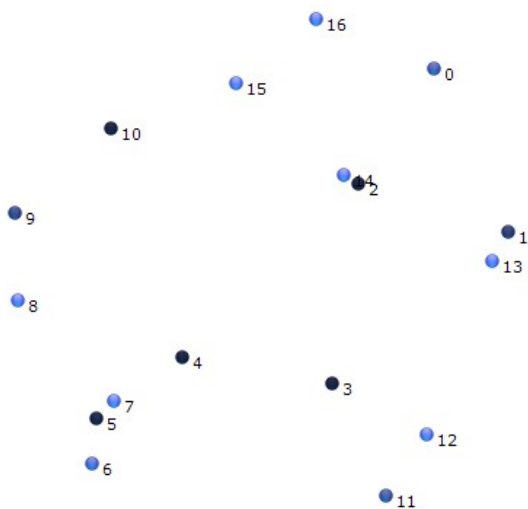
17. Pav.: RPM, vientisas fonas



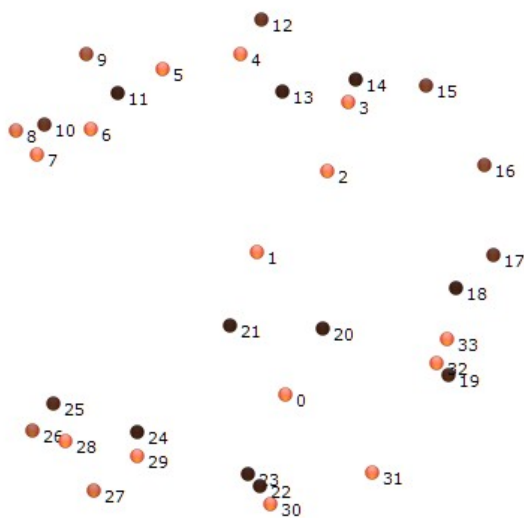
18. Pav.: RPM, nevientisas fonas

Abiem (17ir 18) atvejais, santykines perspektyvos metodai musu uzdaviniui nera tinkamas. Panašumai tarp atvaizdu rasti, ir jų projekcijos paskirstytos grupelėmis. Akivaizdu, kad algoritmas tinka rasti lokaliems panašumams tarp daugdarų.

4.1.4. Manifold RPM



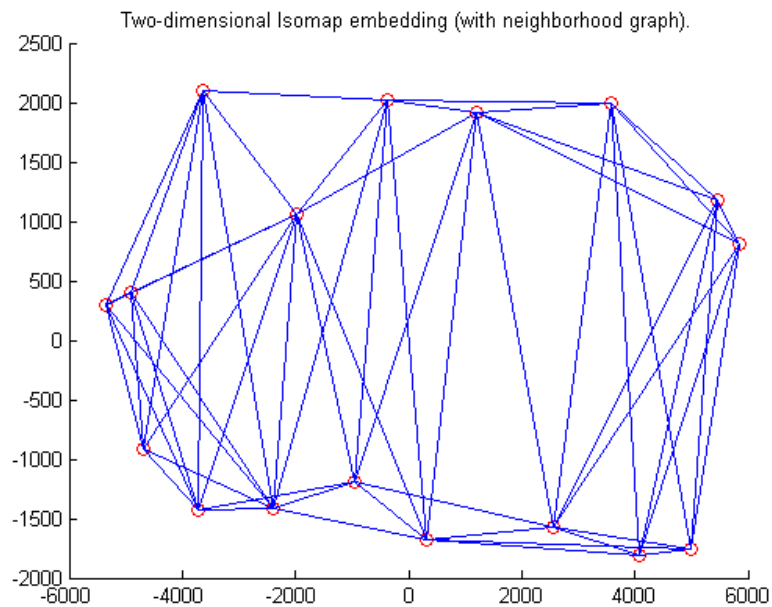
19. Pav.: Manifold RPM, vientisas fonas



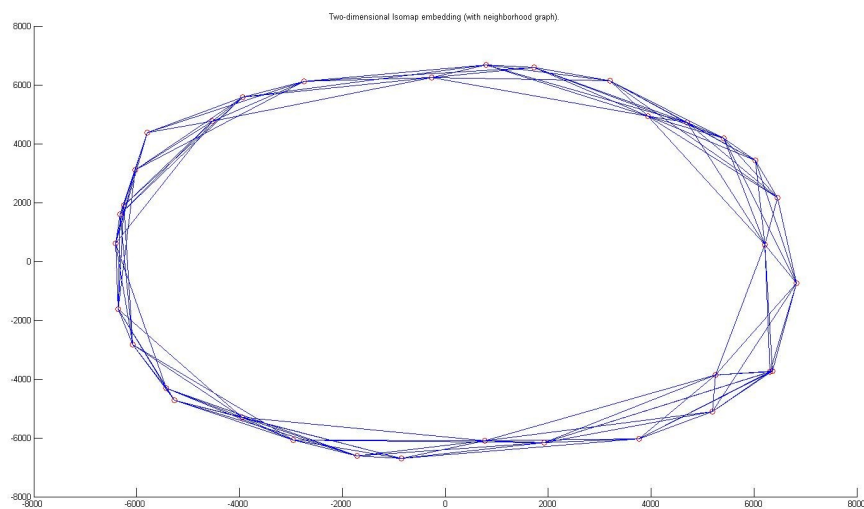
20. Pav.: Manifold RPM, nevientisas fonas

Daugdarų santykinės perspektyvos metodus rezultatus sugrupavo kitaip, nei paprastas RPM, bet vizualiai spręsti apie judesį vistiek sunku. Rezultatai matomi 19 ir 20 paveikslėliuose.

4.1.5. Isomap



21. Pav.: Isomap, vientisas fonas



22. Pav.: Isomap, nevientisas fonas

ISOMAP algoritmas judesį elipse atvaizduoja gerai.

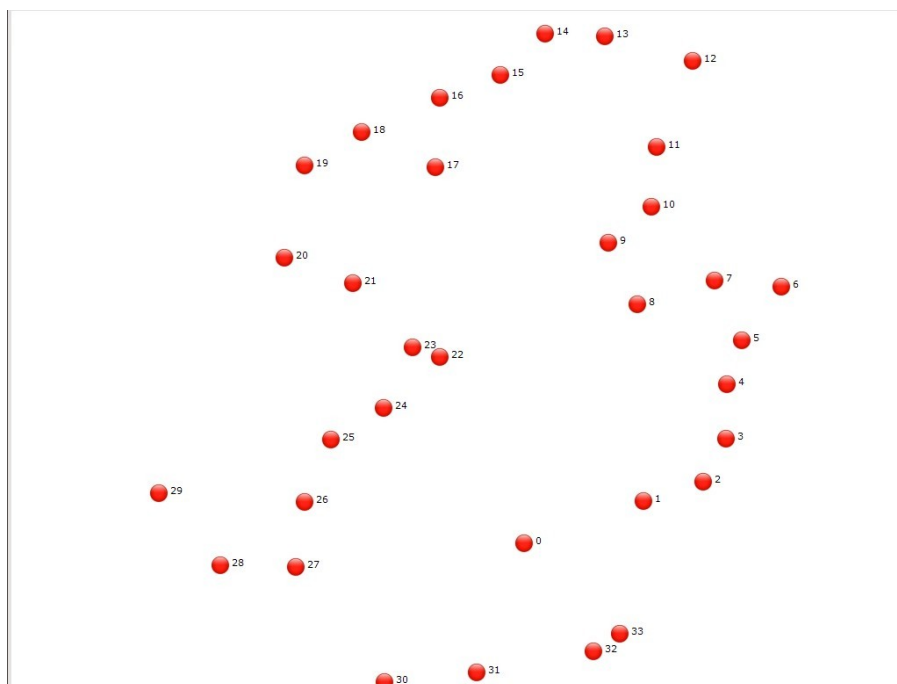
Pirmuoju (21) atveju įtakos rezultato nedideliame netolygumui matyt turėjo mažas atvaizdų kiekis.

Antruoju (22) atveju rezultatai atvaizduoti puikiai.

4.1.6. CCA



23. Pav.: CCA, vientisas fonas

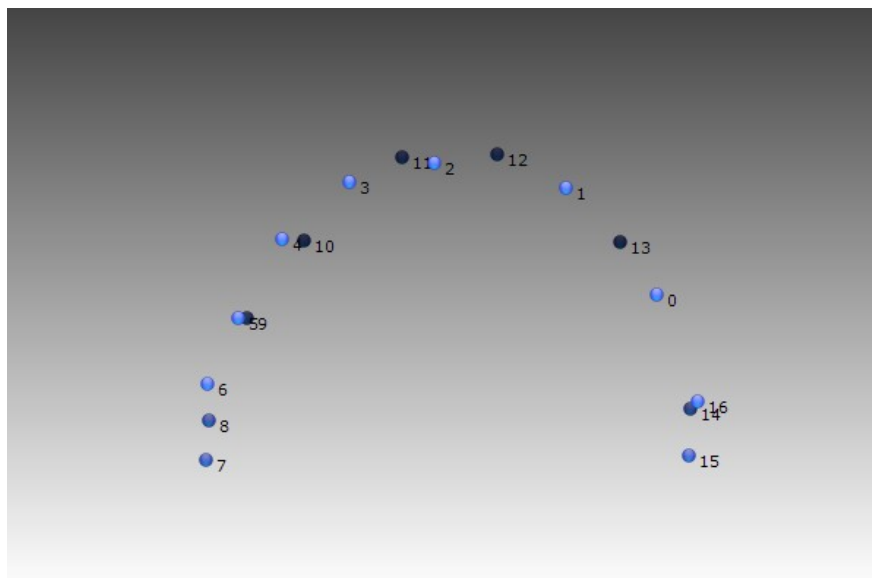


24. Pav.: CCA, nevientisas fonas

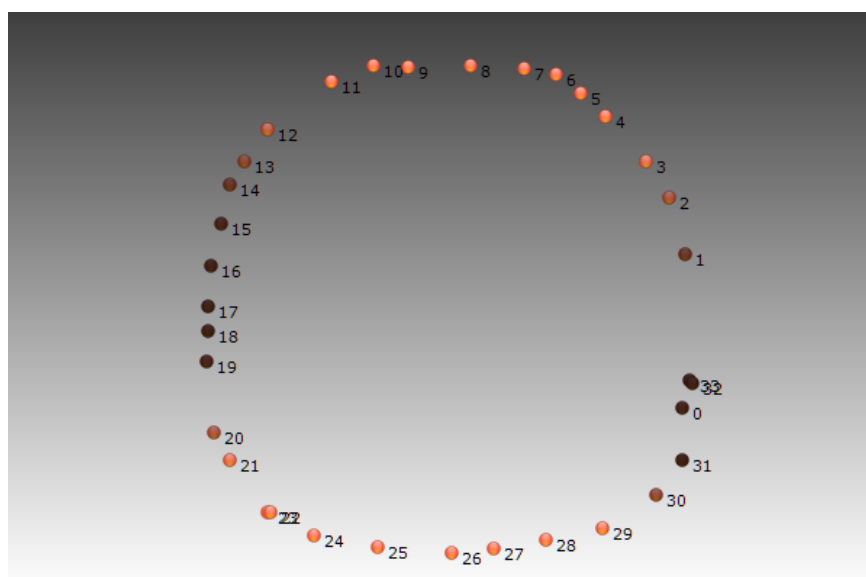
Pirmuoju (23) atveju mažas atvaizdų kiekis ir mažesnė besikeičianti atvaizdo dalis taipogi turėjo įtakos ne visiškai patenkinamiems rezultatams gauti, bet iš akies galima išvesti kreivę, pagal kurią atvaizdų projekcijos išrikiuotos.

Antrojo (24) atvejo rezultatas geresnis, bet neypatingai tinkamas – kreivė beveik atitinka apskritimą, trukdo tik jo prastas taisyklingumas.

4.1.7. PCA



25. Pav.: PCA, vientisas fonas



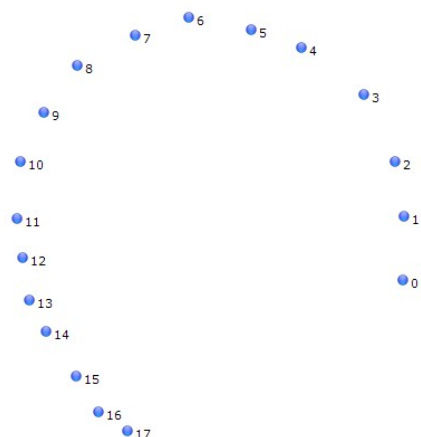
26. Pav.: PCA, nevientisas fonas

Pirmas (25) atvejis šio algoritmo pagalba buvo atvaizduotas netinkamai – judesys žvilgsniui esant ir viršuje, ir apačioje buvo projektuotas vienodai, judesio tipo atpažinti neįmanoma. Tai turbūt sąlygoje mažas judančio objekto dydis.

Antru (26) atveju demonstruojamas kitoks rezultatas – judesys atpažintas labai gerai.

4.2. Horizontalus ir vertikalus judėjimas

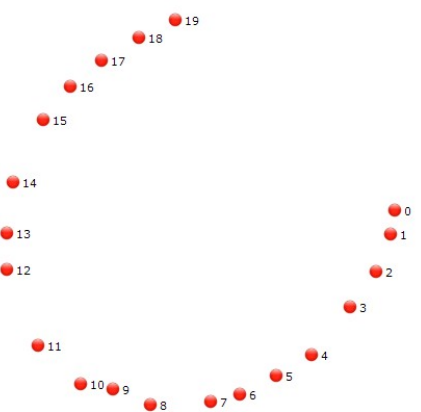
4.2.1. SMACOF



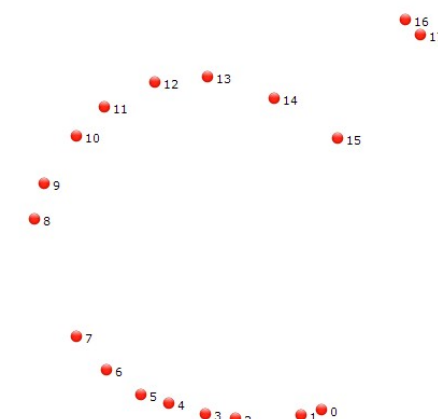
27. Pav.: SMACOF, horizontalus judesys, vientisas fonas



28. Pav. SMACOF, vertikalus judesys, vientisas fonas



29. Pav. SMACOF, horizontalus judesys, nevientisas fonas



30. Pav. SMACOF, vertikalus judesys, nevientisas fonas

Šio algoritmo rezultatai vidutiniški.

Fono įtaka rezultatams beveik nežymi – horizontalus judesys su vientisu fonu (27) ir su nevientisu fonu (28) nustatytas gana tiksliai.

Vertikalus judesys nustatytas prasčiau, judesys vientisu fonu (29) apibrėžtas netolygiai, nevientisu (30 – tolygiai, bet 16 ir 17 projekcijos išsiskiria iš kreivės.

Problema ta, kad projekcijos atvaizduotos į kreives.

4.2.2. Sammon



31. Pav.: Sammon, horizontalus judesys, vientis fonas

32. Pav.: Sammon, vertikalus judesys, vientis fonas



33. Pav. Sammon, horizontalus judesys, nevientis fonas

34. Pav. Sammon, vertikalus judesys, nevientis fonas

Sammon'o algoritmo rezultatai panašūs į SMACOF.

Horizontalūs judesiai su skirtingais fonais apibrėžti panašiai (31 ir 33).

Vertikalūs judesiai irgi beveik atitinka – vientiso fono (32) rezultatas prastas, nevientisame fone atlikto judesio (34) projekcijos atvaizdavime irgi išsiskiria 16 ir 17 projekcijos.

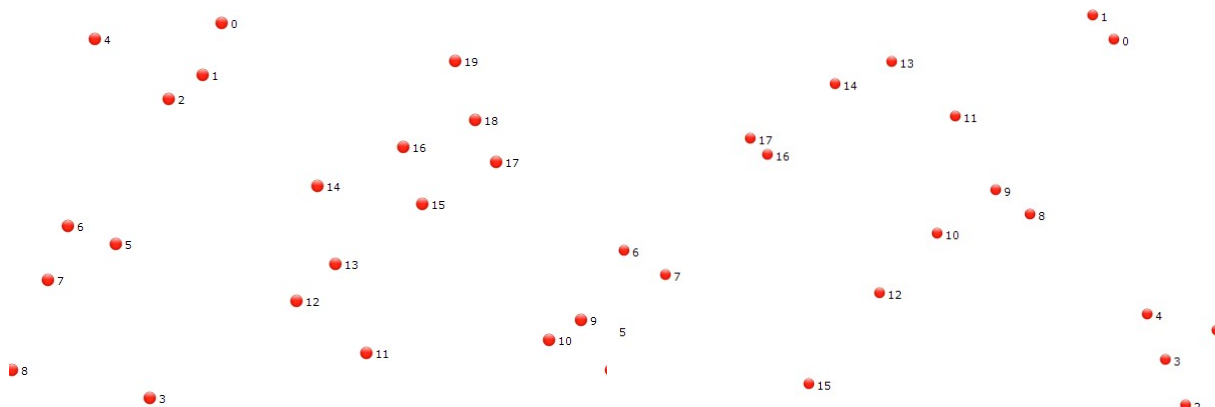
Kreivės gana netolygios ir neatitinka tiesią atkarpą atitinkančio judesio.

4.2.3. RPM



35. Pav.: RPM, horizontalus judesys, vientisas fonas

36. Pav.: RPM, vertikalus judesys, vientisas fonas



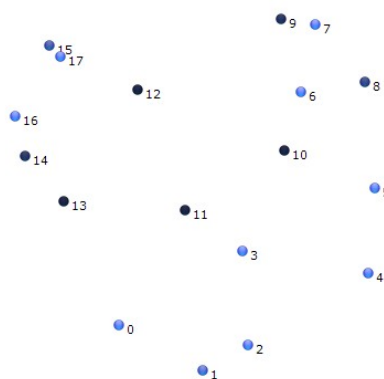
37. Pav. RPM, horizontalus judesys, nevientisas fonas

38. Pav. RPM, horizontalus judesys, nevientisas fonas

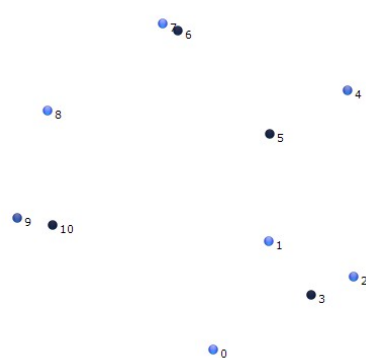
RPM rezultatai (35-38 pav.) tipiški.

Rezultatai sugrupuoti grupelėmis, vizualiai judesio atpažinti neįmanoma.

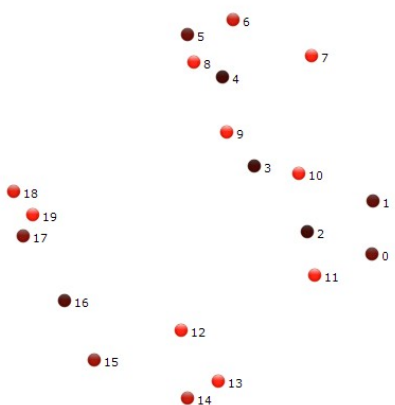
4.2.4. Manifold RPM



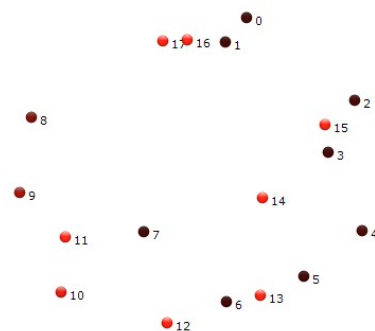
39. Pav.: mRPM, horizontalus judesys, vientisas fonas



40. Pav.: mRPM, vertikalus judesys, vientisas fonas



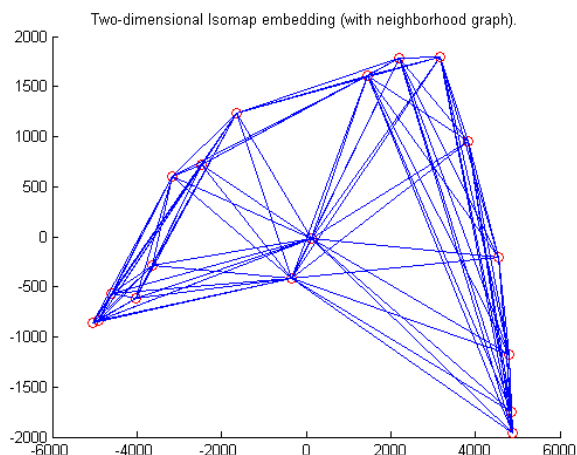
41. Pav. mRPM, horizontalus judesys, nevientisas fonas



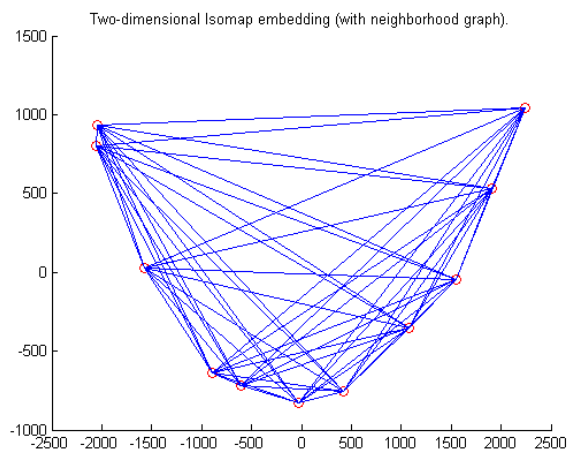
42. Pav. mRPM, vertikalus judesys, nevientisas fonas

Manifold RPM algoritmo rezultatai (39-42 pav.) taipogi neparodo nieko naujo, judesys aiškesnis, nei paprasto RPM algoritmo, bet akivaizdu, kad mūsų užduočiai jis nėra tinkamas.

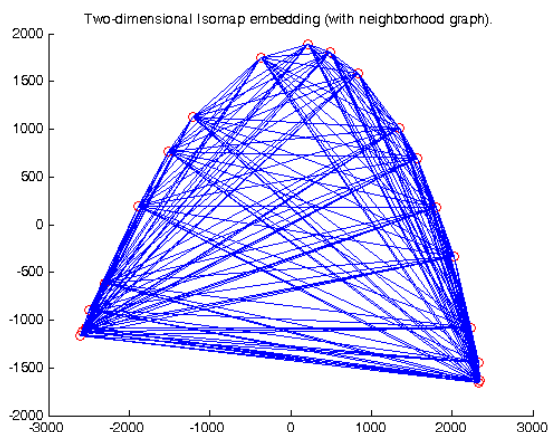
4.2.5. Isomap



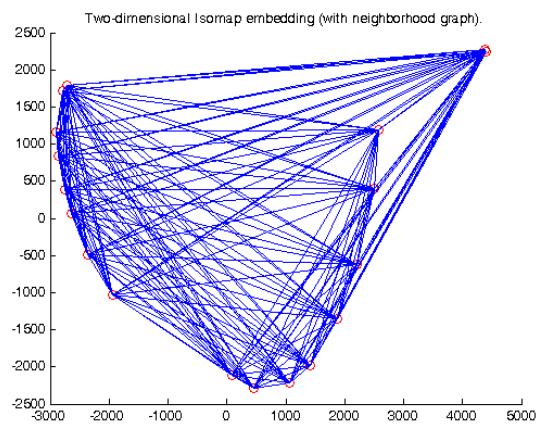
43. Pav.: Isomap, horizontalus judesys, vientisas fonas



44. Pav.: Isomap, vertikalus judesys, vientisas fonas



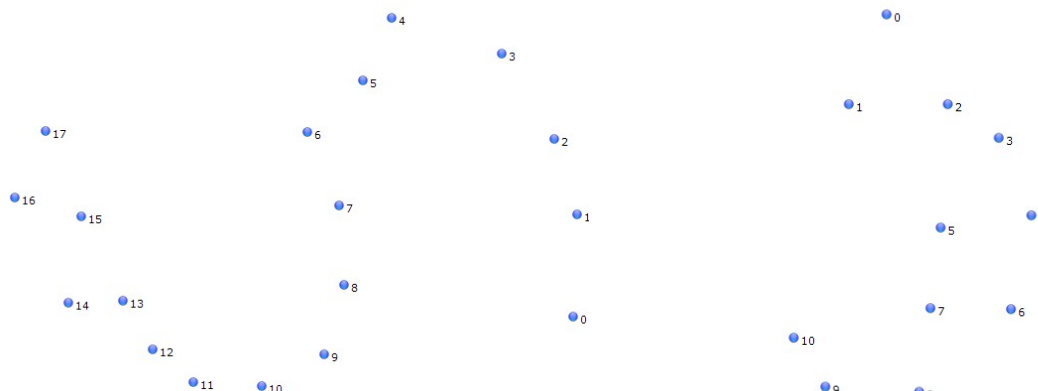
45. Pav. Isomap, horizontalus judesys, nevientisas fonas



46. Pav. Isomap, vertikalus judesys, nevientisas fonas

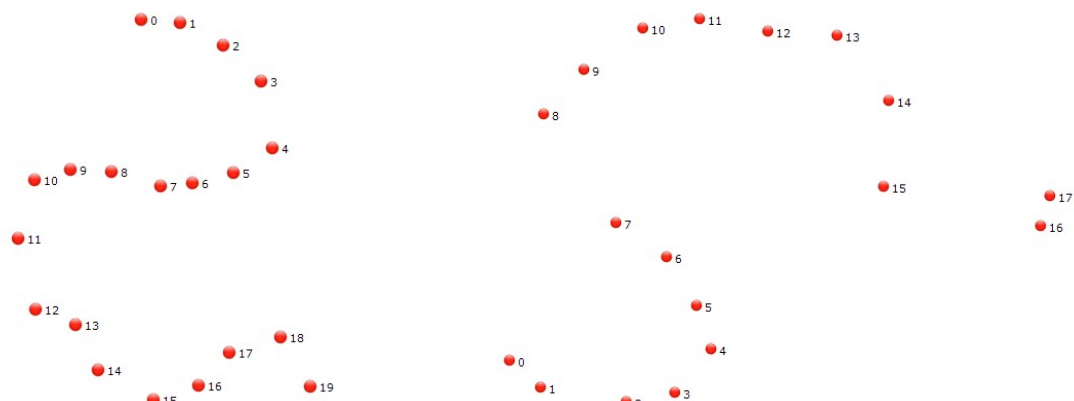
Isomap algoritmo rezultatai neblogi, judesiai atvaizduoti į kreives, bet reikalinga aptikti optimalius veikimo parametrus, nes egzistuoja iškrentančių iš bendro konteksto projekcijų (43 ir 46). 44 ir 45 projekcijų išdėstymai gana tolygūs.

4.2.6. CCA



47. Pav.: CCA, horizontalus judesys, vientisas fonas

48. Pav.: CCA, vertikalus judesys, vientisas fonas



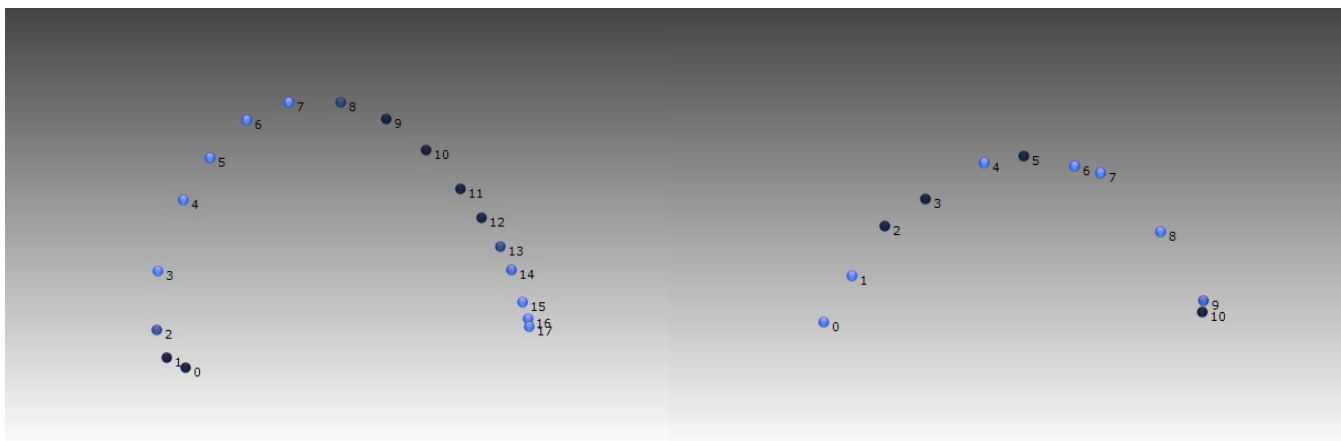
49. Pav. CCA, horizontalus judesys, nevientisas fonas

50. Pav. CCA, vertikalus judesys, nevientisas fonas

CCA algoritmas tiesaus judesio projekcijas išdėstė serpentinai – vizualiai judesio tolygumas matomas, bet jo tipą nustatyti sunku (47,49, 50 pav.).

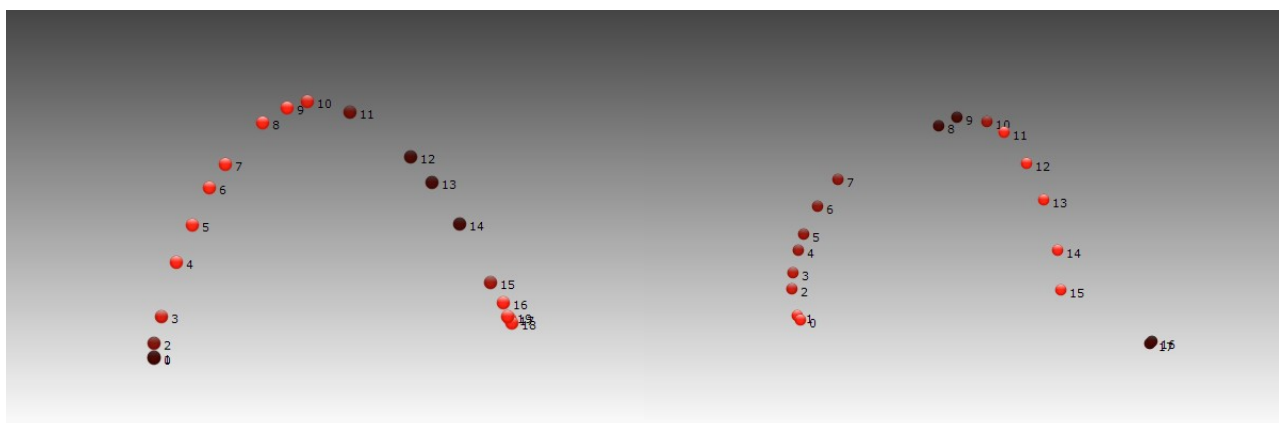
Abejonių kelia 48 paveiksle pavaizduotas grafinis vaizdas – galbūt dėl atvaizdų trūkumo, tolydumo tarp judesio projekcijų visiškai negalima pamatyti.

4.2.7. PCA



51. Pav.: PCA, horizontalus judesys, vientisas fonas

52. Pav.: PCA, vertikalus judesys, vientisas fonas



53. Pav. PCA, horizontalus judesys, nevientisas fonas

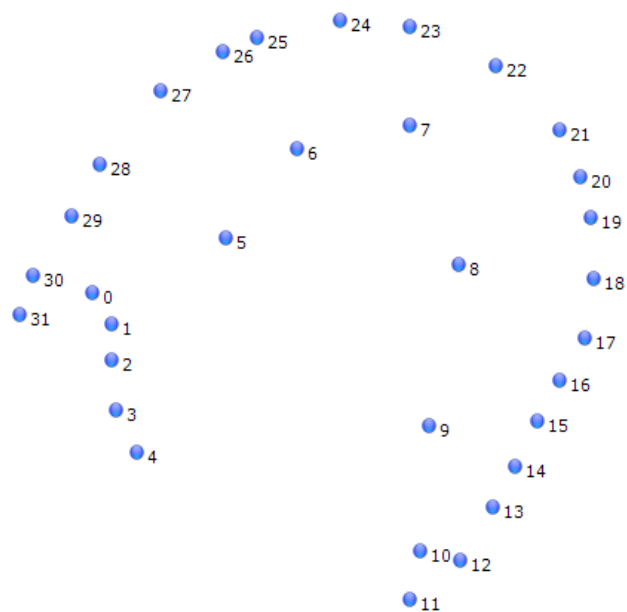
54. Pav. PCA, horizontalus judesys, nevientisas fonas

PCA algoritmas horizontalų (51 ir 53) ir vertikalų (52 ir 54) judesius atvaizdavo gerai, nepriklausomai ir nuo fono, ir nuo judančio objekto dydžio, ir nuo vaizdų kiekio.

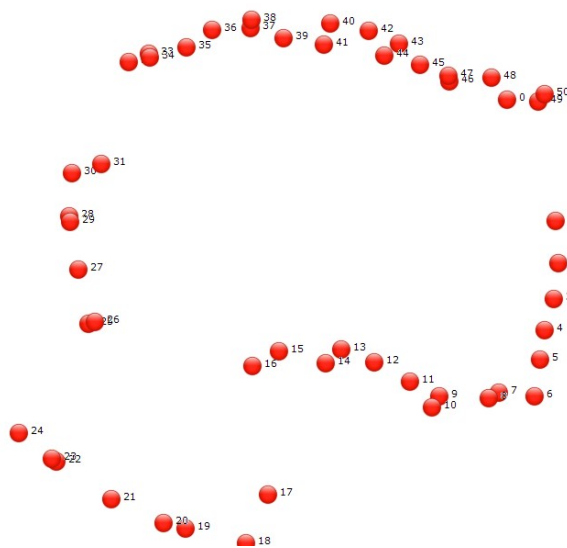
Nerimą dėl aiškumo keltų nebent tarpas tarp 7 ir 8 projekcijų 54 paveiksle.

4.3. Judėjimas kvadratu

4.3.1. SMACOF



55. Pav.: SMACOF, judesys kvadratu, vientisas fonas

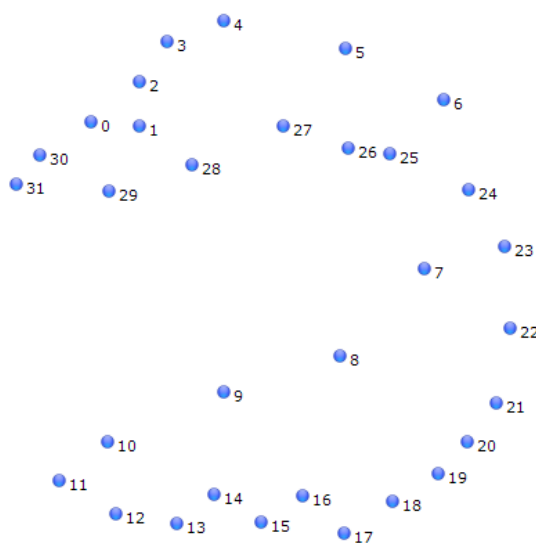


56. Pav.: SMACOF, judesys kvadratu, nevientisas fonas

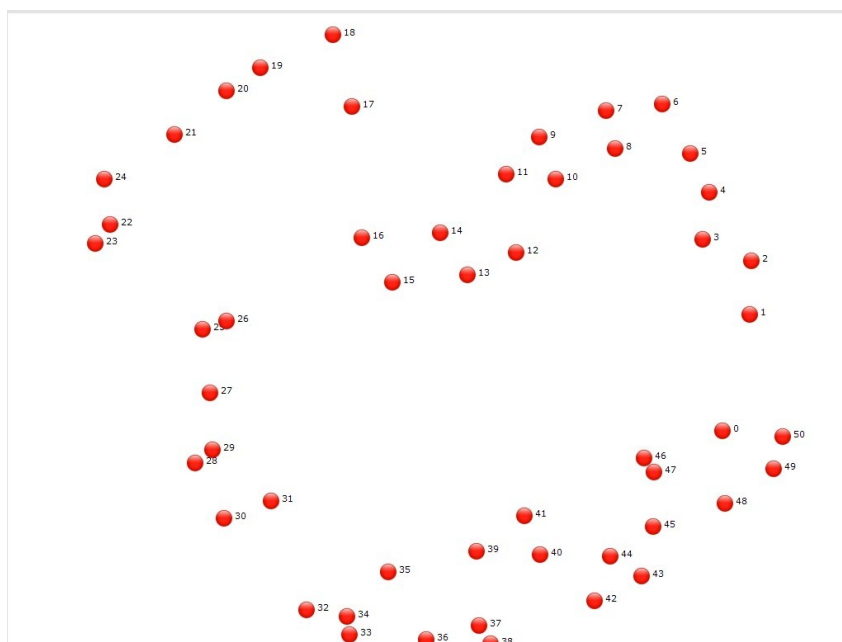
Pirmuoju atveju (55) algoritmas nepateikė vaizdo, iš kurio būtų galima vizualiai spręsti, koks objekto judesys fiksuojamas. Kaip ir daugeliu kitų atvejų, tai galbūt sąlygoja judančio objekto dydis.

Antras atvejis (56) šiek tiek geresnis, bet taipogi tik patenkinamas.

4.3.2. Sammon



57. Pav.: Sammon, judesys kvadratu, vientisas fonas



58. Pav.: SMACOF, judesys kvadratu, nevientisas fonas

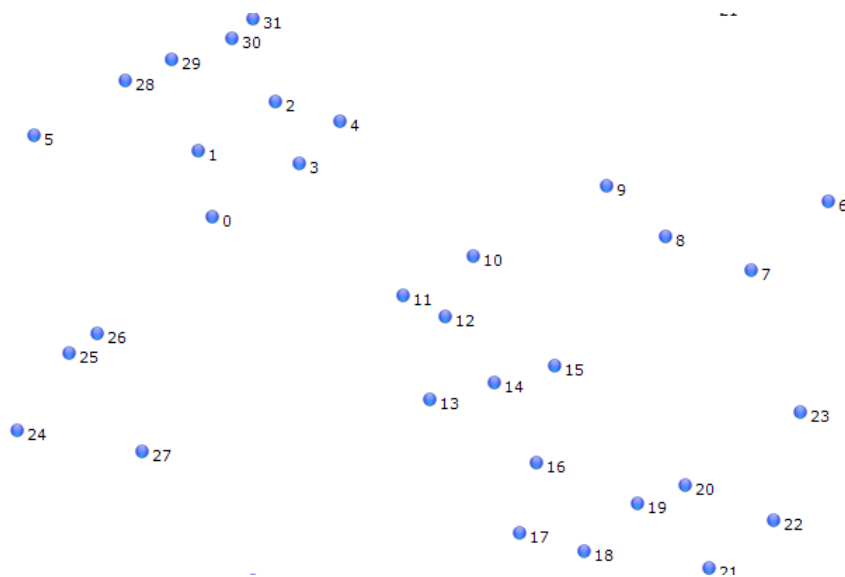
Sammon'o algoritmo rezultatai beveik nesiskiria nuo to pačio algoritmo rezultatų su galvos atvaizdais, kurie juda pagal elipsę.

Pirmuoju atveju (57) rezultatas prastas – 6, 7 ir 24, 25 projekcijos persikerta.

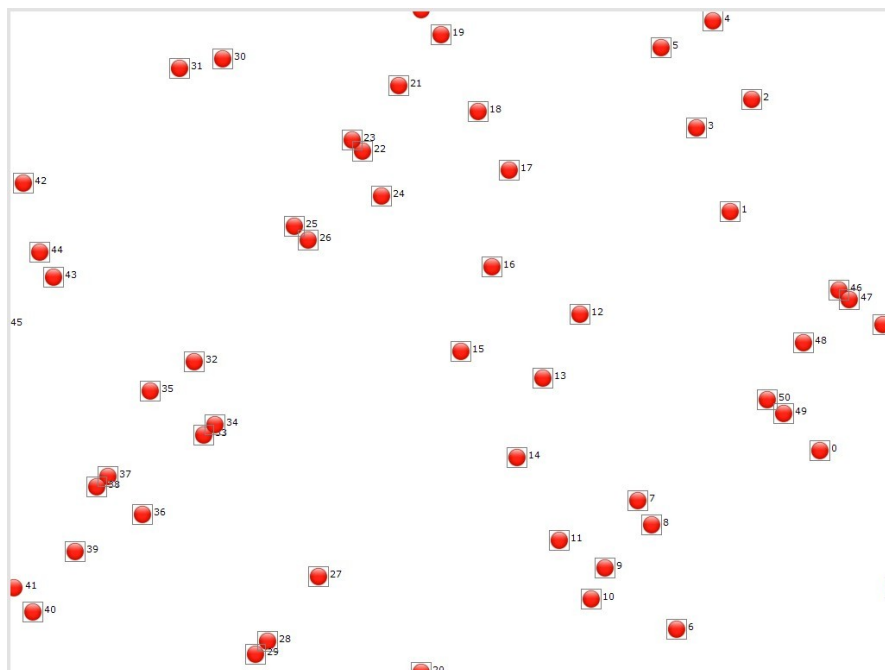
Antrasis atvejis (58) kiek geresnis, atvaizdų projekcijos paskirstytos gana tvarkingai.

Bet, apskritai, šio algoritmo rezultatuose sunku nuspėti, ar atvaizdai sekė keturkampį, ar elipsę – rezultatas nelabai geras.

4.3.3. RPM



59. Pav.: RPM, judesys kvadratu, vientisas fonas

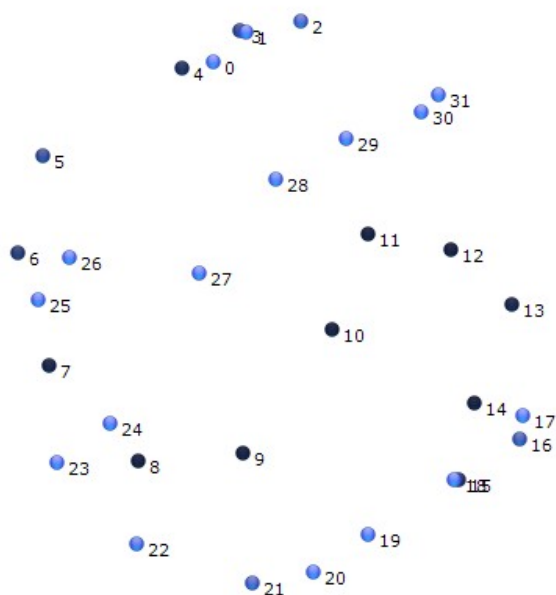


60. Pav.: RPM, judesys kvadratu, nevientisas fonas

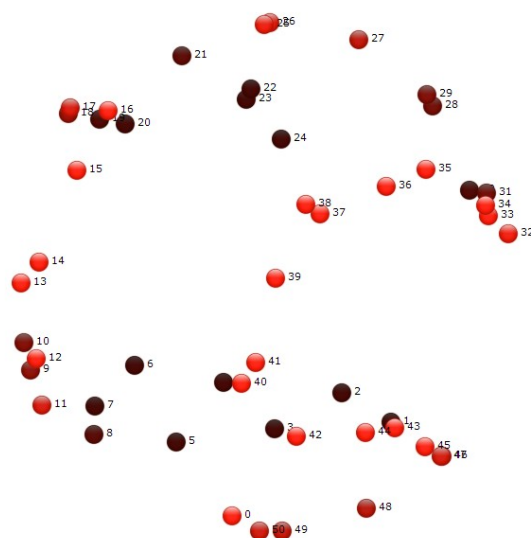
Kaip ir praeitame eksperimente su santykinės perspektyvos metodu, rezultatai beveik tokie pat prasti.

Pirmasis (59) ir antrasis (60) atvejai beveik nesiskiria. Projektijos sugrupuotos grupelėmis pagal panašumą.

4.3.4. Manifold RPM



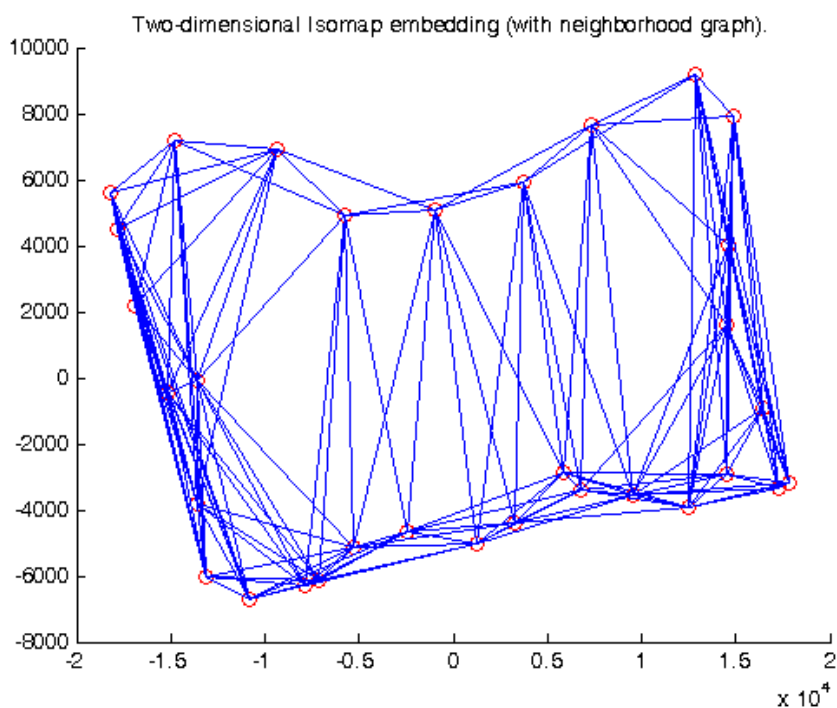
61. Pav.: mRPM, judesys kvadratu, vientisas fonas



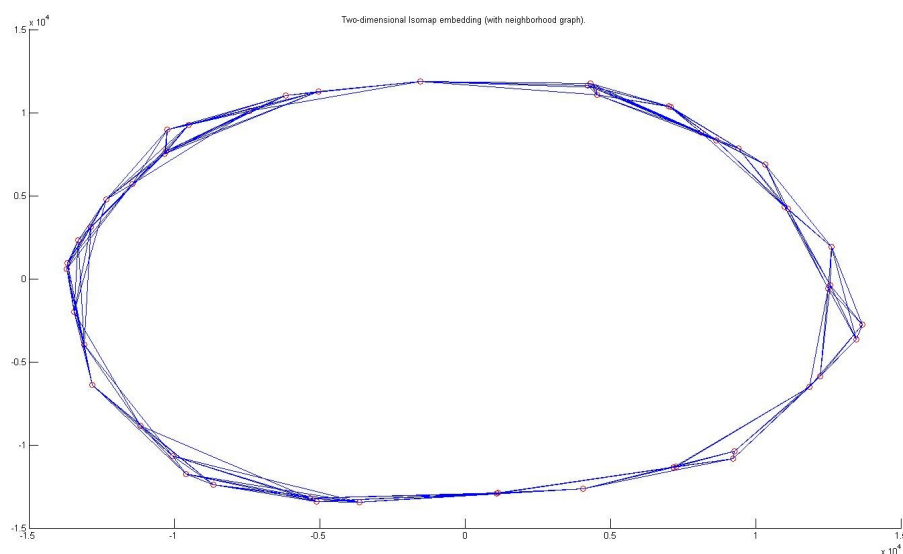
62. Pav.: mRPM, judesys kvadratu, nevientisas fonas

Ir šios bandymo dalies metu MRPM algoritmas mums nieko naujo nepateikė – projekcijos sugrupuotos (61 ir 62), bet vizualinės informacijos, pagal kurią galima nustatyti judėjimo tipą ar kampą, nelabai yra.

4.3.5. Isomap



63. Pav.: Isomap, judesys kvadratu, vientisas fonas



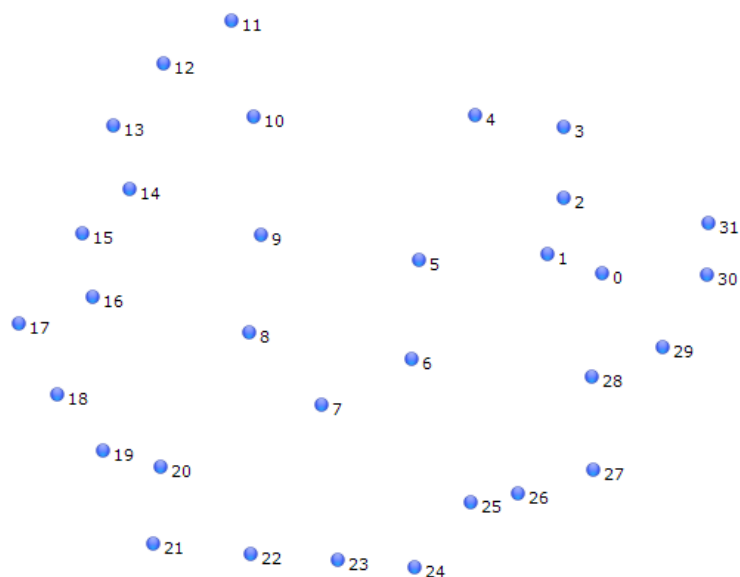
64. Pav.: Isomap, judesys kvadratu, nevientisas fonas

ISOMAP metodas pateikia gan gerus rezultatus.

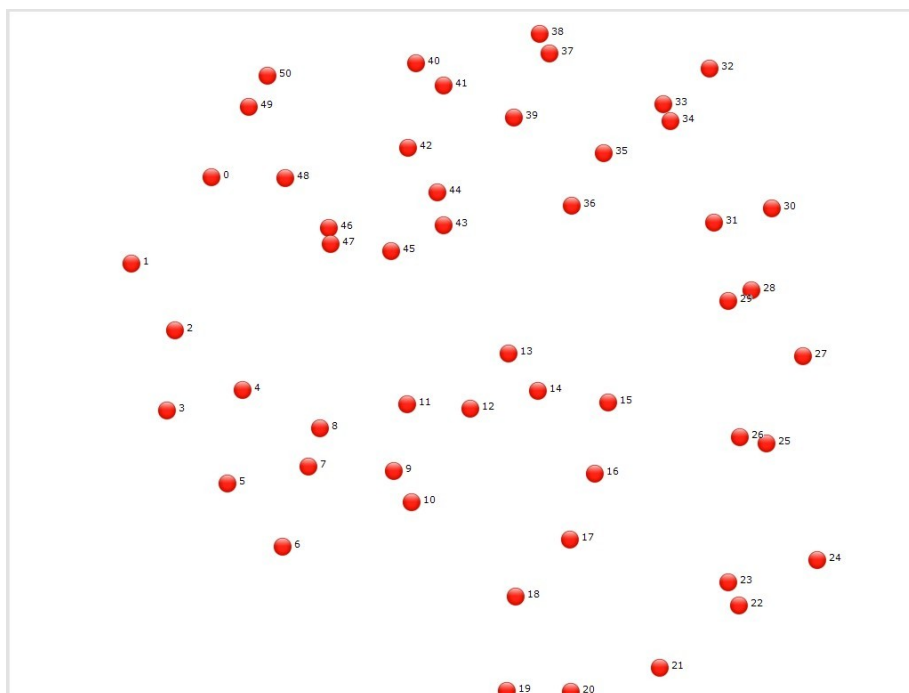
Pirmuoju atveju (63) akivaizdu, kad objektas atvaizduose judėjo stačiakampiu. Vienintelis minusas buvo tas, kad skaičiuojant reikėjo imti didesnę kaimynų skaičių, tikslesniam rezultatui gauti.

Antro atvejo (64) rezultatas šiek tiek prastesnis, matyt dėl didelio objektų kiekio fone, ne taip aiškiai algoritmo randamas judesys.

4.3.6. CCA



65. Pav.: CCA, judesys kvadratu, vientisas fonas

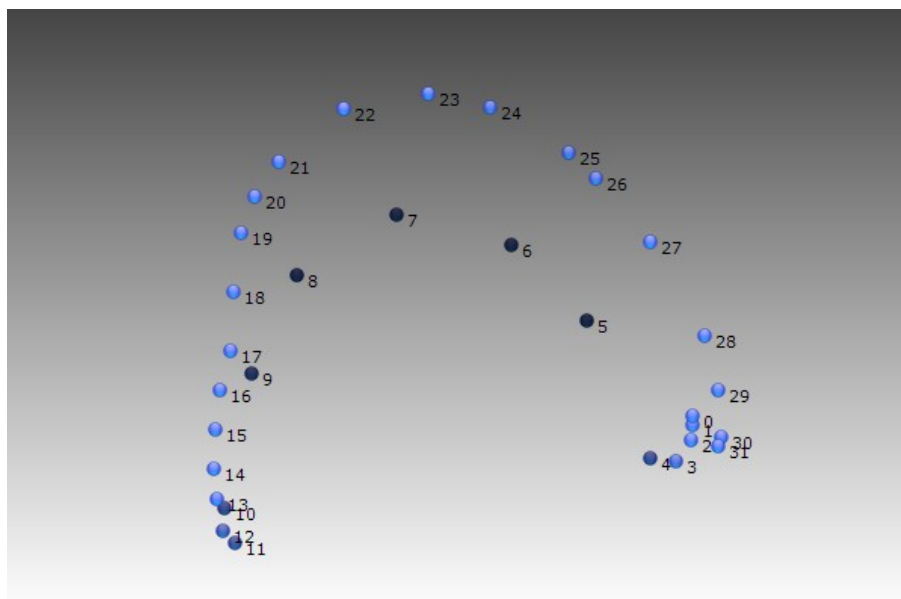


66. Pav.: CCA, judesys kvadratu, nevientisas fonas

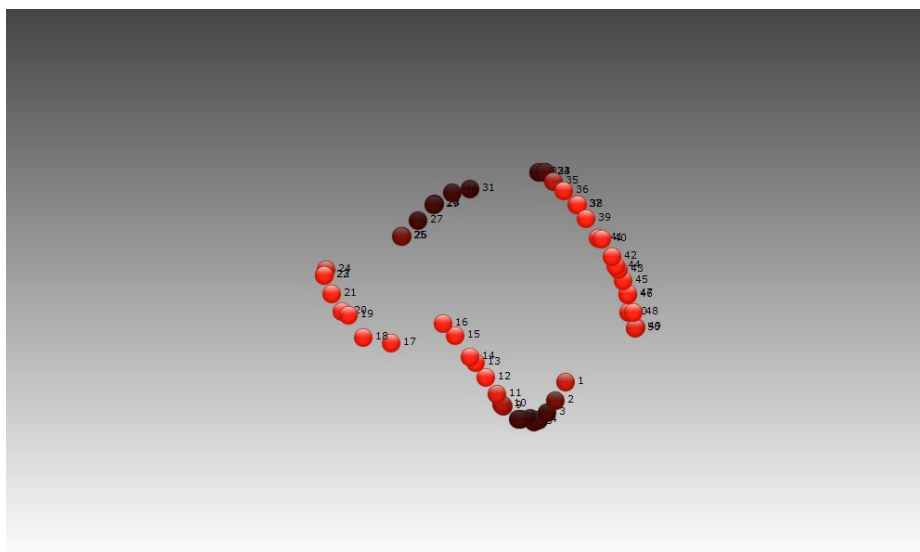
Kreivinių komponentių metodo rezultatai mažai skiriasi nuo to pačio metodo rezultatų su objekto judėjimu elipse.

Ir pirmu (65), ir antru (66) atvejais galima nustatyti, kad objektas judėjo uždara kreive, bet kokia ta kreivė buvo, vizualiai nėra aišku.

4.3.7. PCA



67. Pav.: PCA, judesys kvadratu, vientisas fonas



68.. Pav.: PCA, judesys kvadratu, nevientisas fonas

Šio metodo rezultatai įdomūs.

Pirmu atveju (67) rezultatas visiškai nepatenkinamas, rezultato vizualiai nustatyti negalima.

Antru atveju (68) algoritmas atskyrė visus keturi judesius palei visas ekrano kraštines, ir juos atskyrė.

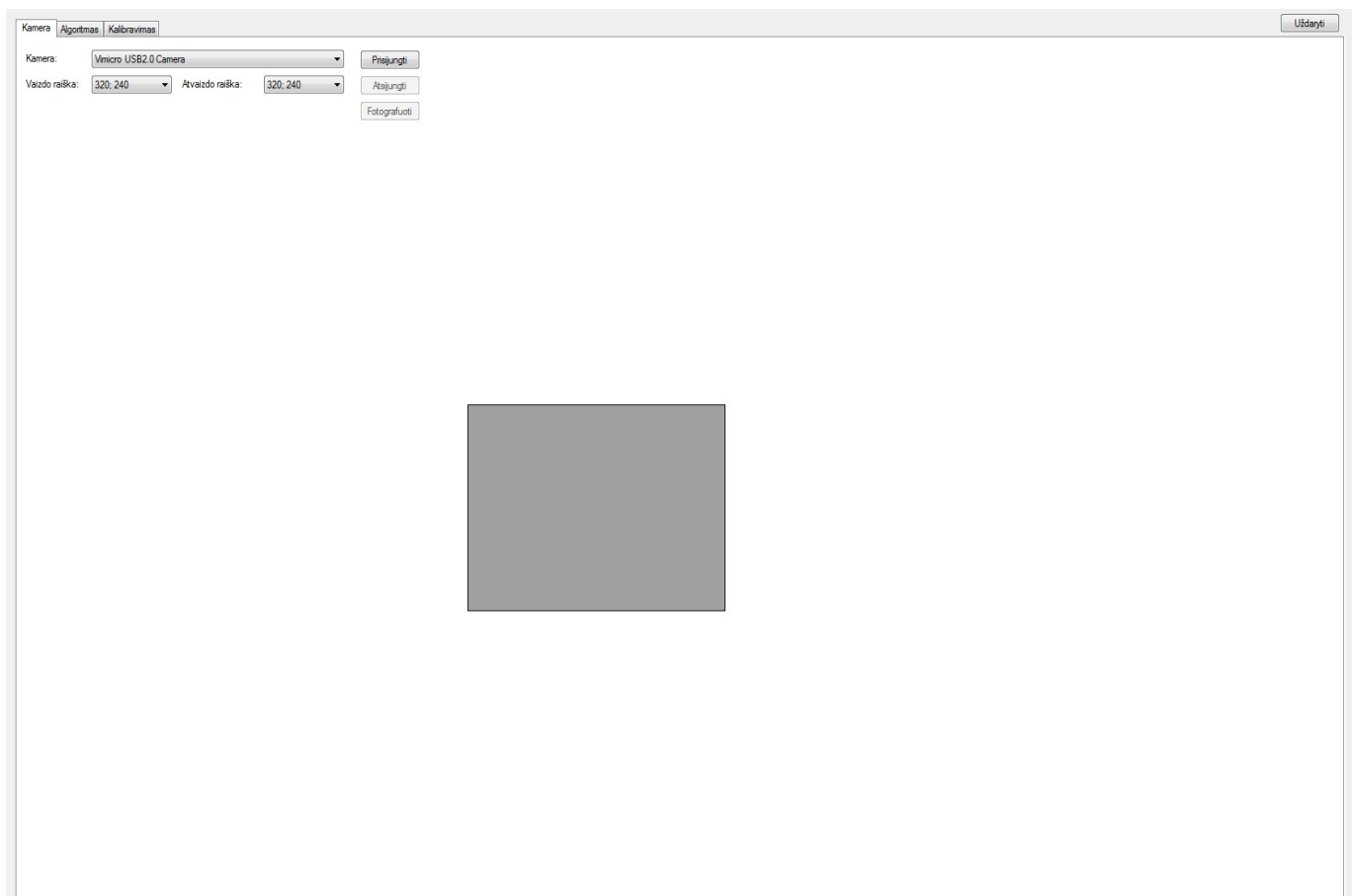
5. Praktinis Isomap algoritmą panaudojančios programos įgyvendinimas

5.1. Programos sandara

Programa susideda iš 3 dalių:

1. Kameros valdymo skiltis (pasirenkama kamera, nustatoma vaizdo raiška ir patikrinamas kameros rodomas vaizdas) (69 pav.);
2. Algoritmo valdymo skiltis (pasirenkami algoritmo veikimo parametrai);
3. Kalibravimo skiltis (atliekamas vaizdų, gautų iš kameros, išsaugojimas) (70 pav.).

Vaizdus išsaugojus, paleidžiamas Isomap algoritmas ir galima nagrinėti rezultatus.



69. pav. Valdymo skiltis

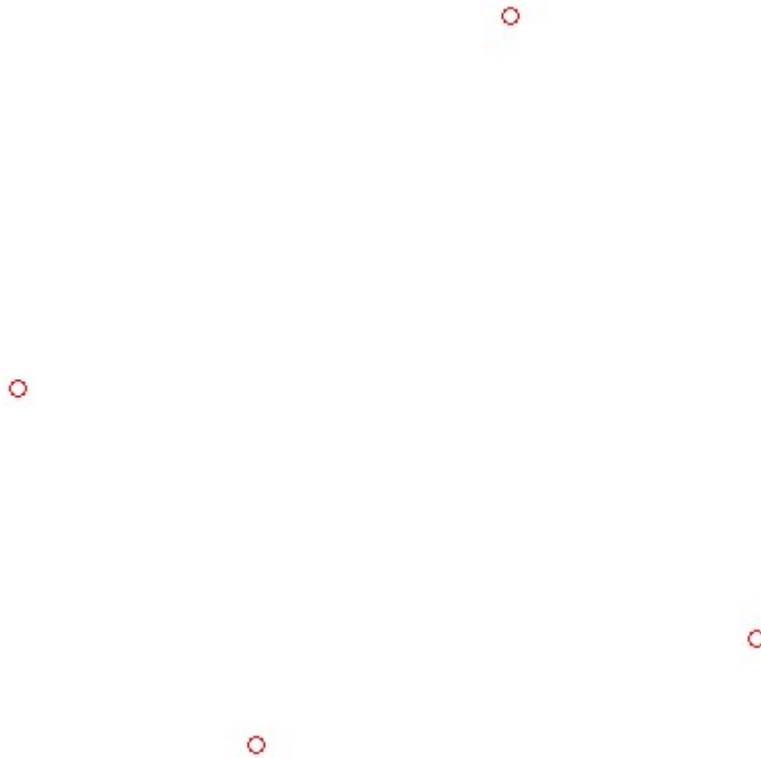


70 pav. Kalibravimo skiltis.

5.2. Programos algoritmas

1. Sukuriami galvos atvaizdai.
2. Visi atvaizdai paverčiami nespaltovais.
3. Sudedami į tokio pavidalo masyvą $N \times 76800$ (N – atvaizdų kiekis, 76800 – pikselių kiekis atvaizde = $320 * 240$).
4. Iš minėto masyvo gaunama Euklidinių atstumų matrica $N \times N$
5. Iš Euklidinių atstumų matricos gaunamas kaimynų grafas.
6. Floido algoritmu apskaičiuojami trumpiausieji atstumai tarp taškų.
7. Daugiamačių skalių metodu taškai sudedami į mažesnės dimensijos erdvę.

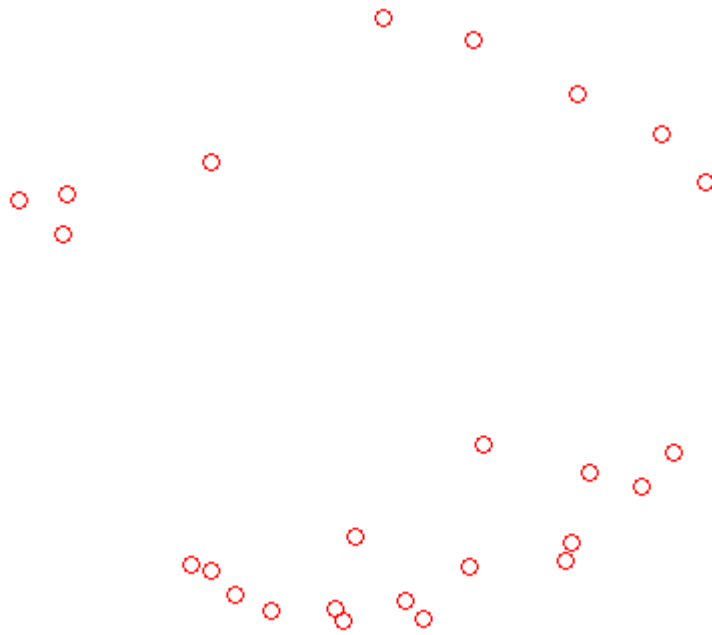
5.3. Eksperimentai



71 pav. 4 kampai

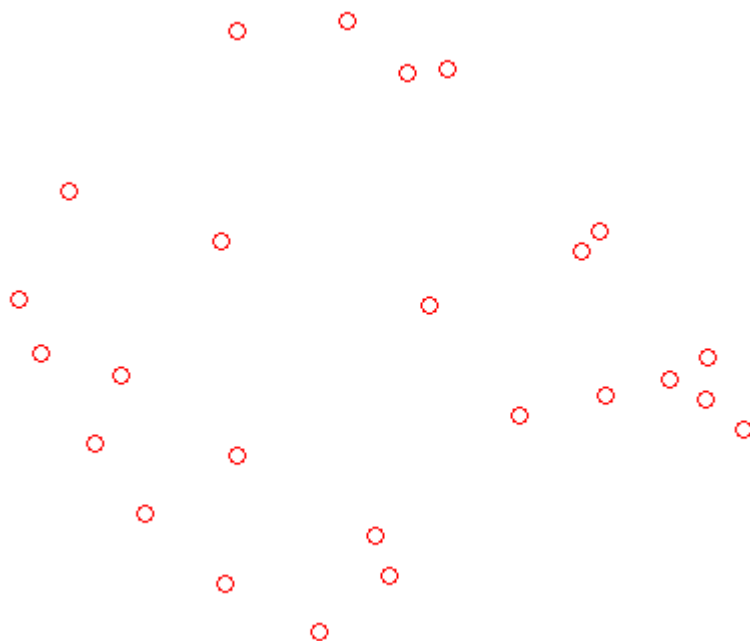
Atvaizdas, žiūrint į keturis ekrano kampus (71 pav.).

Apačioje matome, kaip palenktos galvos (lyginant su ekrano viršuje esančia kamera) mažiau apšviesta veido dalis duoda daug mažiau įtakos rezultatams, nei viršutinė veido dalis.



72 pav. 25 taškai

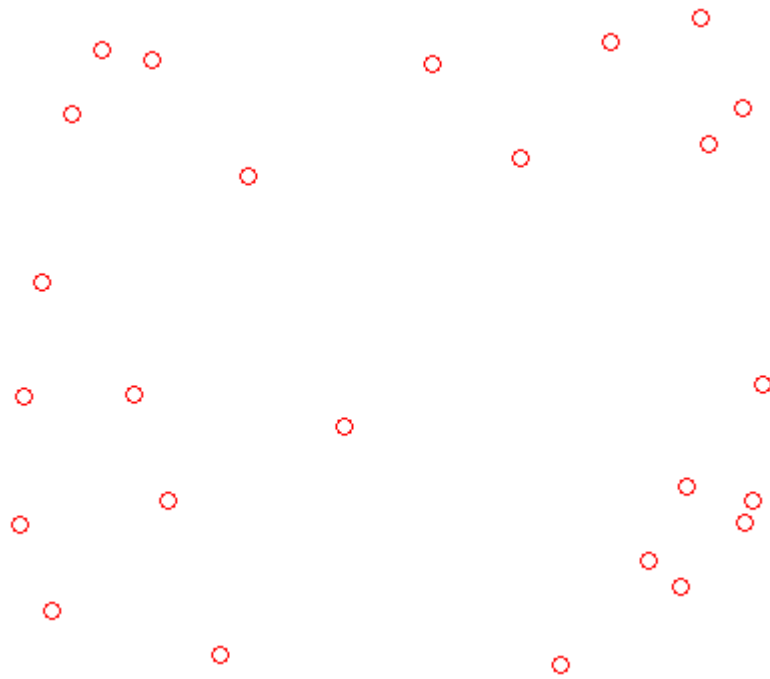
Rezultatas, kai paimti atvaizdai spirale apžiūrint 25 ekrano taškus (72 pav.), dėl šviesos trūkumo rezultatas visiškai nepatenkinamas.



73 pav. 25 taškai.

Rezultatas, kai paimti tokie pat 25 atvaizdai, tik veidą apšvietus (73 pav.).

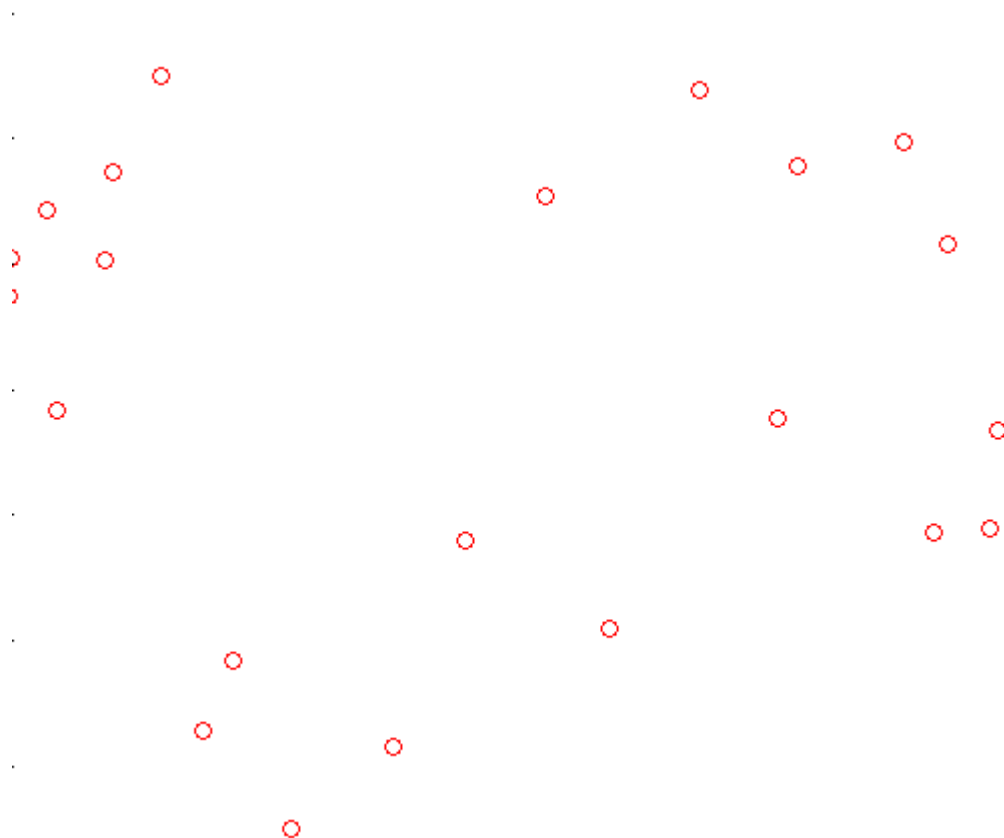
Globali struktūra įžiūrima, rezultatas vidutiniškas.



74 pav. 25 taškai

Rezultatas, kai paimami 25 atvaizdai, apžiūrint ekrano taškus zigzagu (kairėn-žemyn-dešinėn-žemyn ir t.t.) (74 pav.).

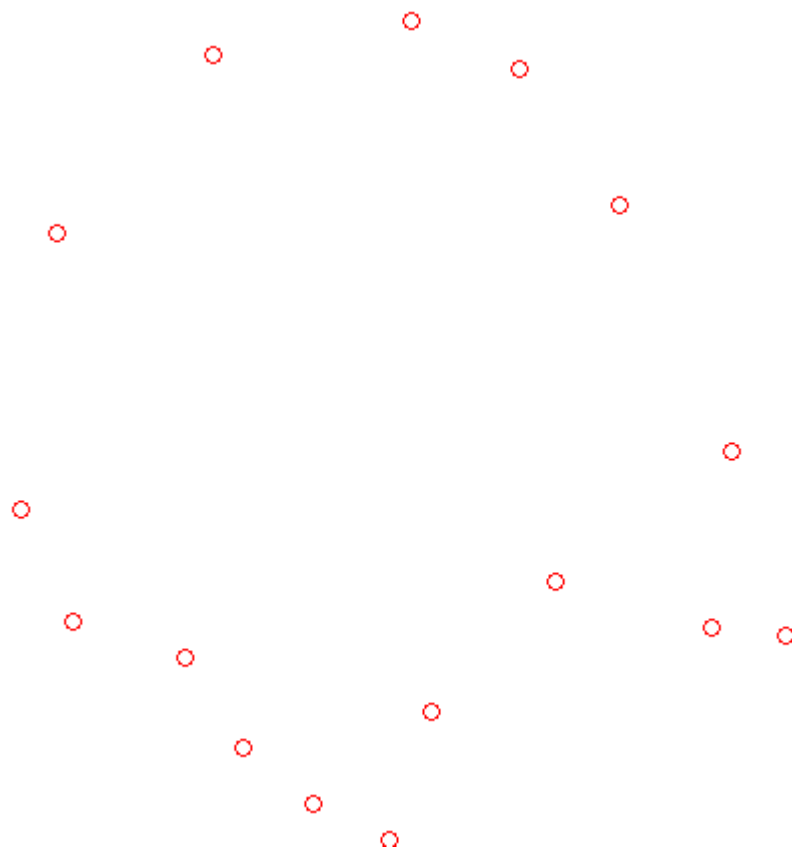
Rezultatas gana aiškus, vėlgi, menčiau matoma ir mažiau apšviesta apatinė veido pusė, palenkus galva išskiriama rezultatuose sunkiau.



75 pav. 25 taškai.

Rezultatas, kai paimami 25 atvaizdai, apžiūrint zigzagu, bet pirmi 3 rezultatai neįtraukiami į skaičiavimus (75 pav.).

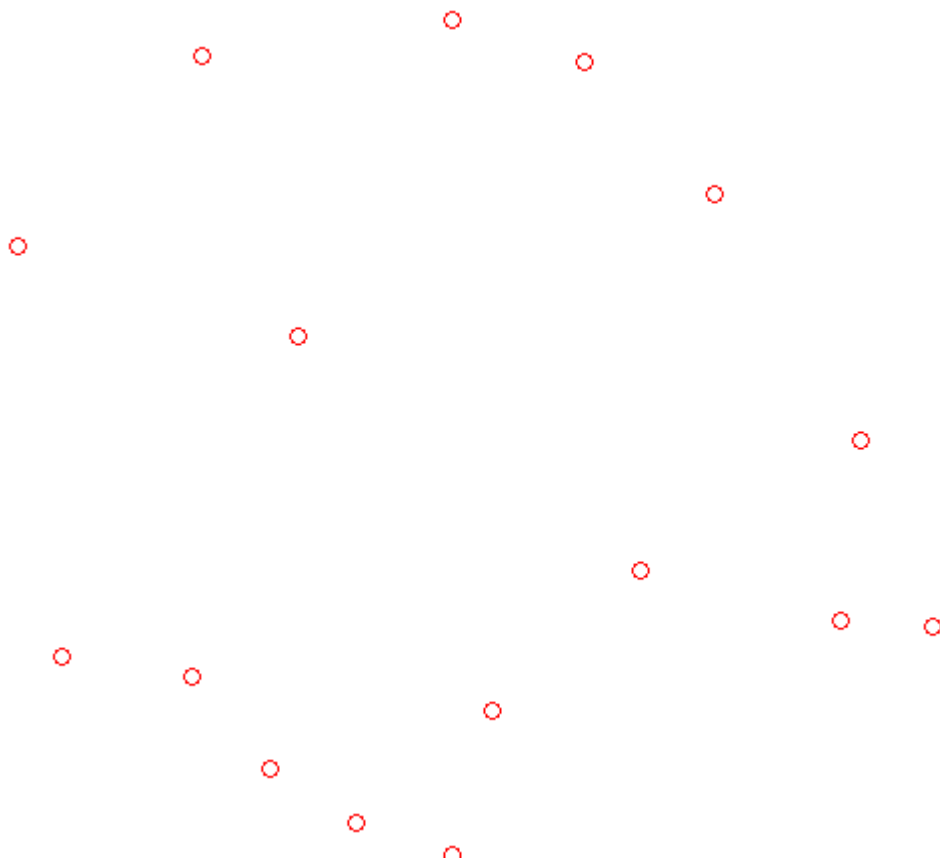
Rezultatas gerokai suprastėjęs.



76 pav. 16 vaizdų, kvadratas

Rezultatas, kai paimami 16 vaizdų, apžiūrint ekraną kvadratu (76 pav.).

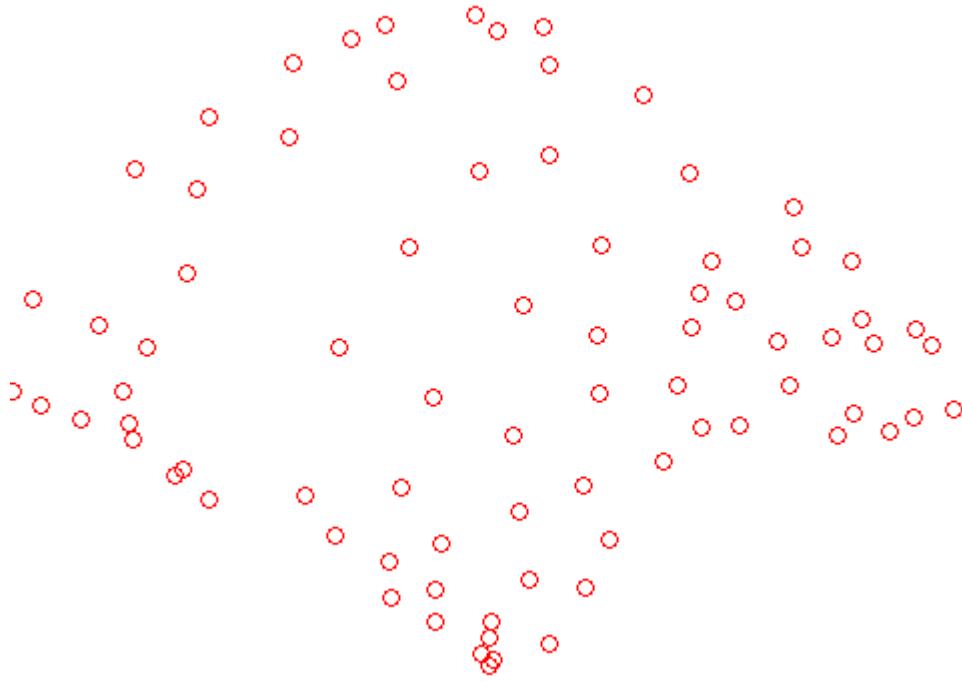
Rezultatas gan geras, šiek tiek iškreiptos viršutinė ir apatinė linijos.



77 pav. 16 vaizdų + centras

Toks pats eksperimentas, kaip ir anksčiau, tik pridėtas atvaizdas, kai žiūrima į ekrano centrą (77 pav.).

Rezultatas geras, nuo praeito nepasikeitęs, atsiranda taškas viduryje.



78 pav. 81 taškas.

Geriausias rezultatas – kai paimami galvos atvaizdai, ekraną apžiūrint zigzagu (78 pav).

Padidintas vaizdų paėmimo tankis 9x9 sektoriai.

Vienintelė mažiau aiški vieta yra apatinis kairysis kampas, kuris mažiau apšviestas iš kairės pusės.

6. Išvados ir rezultatai

Išvados:

- Geriausiai galvos pasukimo eksperimento rezultatus pateikia Isomap metodas.
- Fone tolygumas ar netolygumas (fone esant kažkokiam kiekiui nejudančių daiktų) rezultatams įtakos turi tik kai kuriais atvejais.
- Atvaizdų kiekis turi įtakos rezultatams, didėjant vaizdų kiekiui, tikslumas taipogi didėja.
- Jei judantis objektas užima didesnę atvaizdo dalį, rezultatai būna tikslesni.
- Atvaizdų dydis rezultatams įtakos neturi, sumažinus paimamų vaizdų raišką nuo 640x480 iki 320x240 (dvigubai) rezultatas beveik nekinta.
- Apšvietimas turi didelę įtaką galutiniam algoritmo darbo rezultatui, esant nepakankamam apšvietimui, arba apšviečiant tik vieną veido pusę, gaunami visiškai netikslūs rezultatai.
- Atvaizdai, turintys mažesnę kontrastą, rezultate atvaizduojami, kaip panašesni.
- Tiksliausias rezultatas pasiekiamas, algoritmo veikimo metu atsižvelgiant į visus daugdaros „kaimynus“.
- Žiūrint į apatinę ekrano dalį, atvaizdus atskirti sunkiau, galbūt tai padėtų išspręsti identiška kamera, fiksuojanti vaizdą nuo ekrano apačios.

Rezultatai:

- Apžvelgti daugiamačių duomenų dimensijos mažinimo metodai:
 - Santykinės perspektyvos metodas;
 - Pagrindinių komponentų analizė;
 - Daugiamatės skalės metodas;
 - metodas atliekant sudėtingos funkcijos mažorizavimą;
 - Trianguliacija;
 - Sammon'o algoritmas.
- Apžvelgti netiesinių daugdarų atpažinimo metodai:
 - Kreivinių komponentų analizė;
 - ISOMAP;

- Lokaliai tiesinis vaizdavimas;
- Laplaso matricos tikrinių žemėlapių metodas;
- Hesės matricos tikrinių žemėlapių metodas;
- Lokaliųjų liečiamųjų erdvių rikiavimas.
- Sugeneruotas tiriamų duomenų rinkinys.
- Parinkti objekto judesio būdai.
- Parinkti daugiamačių duomenų vizualizavimo metodai.
- Eksperimentai su minėtais metodais atlikti.
- Programiškai įgyvendinta aplikacija, atliekanti eksperimentą analizuoti vaizdus Isomap metodu.
- Apibrėžti programos darbo rezultatai.

7. Rekomendacijos

Norint algoritmą panaudoti praktiniam tikslui (kursoriaus ar atitinkamo žymeklio valdymui ar apskritai galvos pasukimo kampui nustatyti), reikėtų algoritmą atlikti su kalibravimo duomenų rinkiniu, pridėjus realiu laiku paimto atvaizdo duomenis.

Galima būtų algoritmo rezultatus išbandyti imant vaizdą iškart iš dviejų kamerų, taip išsispręstų problema, kai dėl aukščiausioje ekrano vietoje esančios kameros, sunkiai matoma apatinė veido dalis.

Realizuojant praktiškai veikiančią programą galvos padėčiai nustatyti, būtų reikalingas galbūt dar didesnis tikslumas, nei 81 vaizdo sektorius, nors tai gali didinti klaidų kiekį, reikalinga apmąstyti ir aiškiai išsiskiriančius tarpusavyje ekrane esančius objektu (ar tai būtų meniu punktai, ar ikonos ar dar kas nors.)

8. Literatūros sąrašas

- [BG97] Borg, I., Groenen, P. Modern Multidimensional Scaling: theory and applications, 1997
- [Sam69] Sammon, J. W. A nonlinear mapping for data structure analysis. 1969
- [Li04] James Xinzhi Li. Visualization of high-dimensional data with relational perspective map. 2004
- [Yan02] Yang, M. Face recognition using extended isomap. 2002
- [LLDV00] John Aldo Lee, Amaury Lendasse, Nicolas Donckers, Michel Verleysen, A Robust Nonlinear Projection Method. 2000
- [Jo89] Jolliffe, I. T. Principal Component Analysis. 1989
- [AT07] Adler, R. J., Taylor, J. E. Random Fields and Geometry. 2007
- [MBT05] Mekuz, N.; Bauckhage, C.; Tsotsos, J. K. Face recognition with weighted locally linear embedding. 2005
- [ZZ06] Zhu, L., Zhu, S. A. Face recognition based on extended locally linear embedding. 2006
- [ZZL05] Zhao, Q., Zhang, D., Lu, H. Supervised LLE in ICA space for facial expression recognition. 2005
- [HKP02] Hadid, A., Kouropteva, O., Pietikäinen, M. Unsupervised learning using locally linear embedding: experiments with face pose analysis. 2002
- [Sac01] Sachinopoulou, A. Multidimensional Visualization. 2001
- [LSB77] Lee, R. C., Slagle, J. R., Blum, H. A triangulation method for the sequential mapping of points from n-space to two-space. 1997
- [Yan02] Yang, M. Face recognition using extended isomap. 2002
- [BN01] Mikhail Belkin, Partha Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, Advances in Neural Information Processing Systems 14. 2001