

ДҮРСНИЙ ИЖИЛ ТӨСТЭЙ ХЭМЖИГДЭХҮҮНИЙГ ТОДОРХОЙЛОХАРГАЧЛАЛ

Г.Ганчимэг

Е-Нээлттэй Институт, ШУТИС, Улаанбаатар, Монгол

Е-хаяг: ganaa@must.edu.mn

Хураангуй-Дүрс боловсруулалт нь дохио боловсруулалтын дэд салбар бөгөөд компьютерийн алгоритм ашиглан тоон дүрсүүд дээр боловсруулалт хийдэг. Дүрсүүд нь хоёр эсвэл түүнээс дээш хэмжээс дээр тодорхойлогддог учир олон хэмжээст системийн хэлбэрээр загварчилдаг. Дүрс танилтын технологи нь хурууны хээ, нүүр, машины дугаар таних, хөдөлгөөн дагах, биетийг тоолох болон хянах, автомат зогсоолын систем, соёлын өвийн дүрсүүдийг нөхөн сэргээх гэх мэтолон салбарт хэрэглэгддэг, хурдацтай хөгжиж буй технологи юм. Тус судалгааны ажлаар дүрсний ижил төстэй хэмжигдэхүүнийг тодорхойлох аргачлалуудын талаар өгүүлнэ.

Keywords: Хаусдорфф, эрчим, SIFT, SURF, ХБК

I. ДҮРСНИЙ ИЖИЛТӨСТЭЙ ХЭМЖИГДЭХҮҮН

Дүрс танилт нь зураг болон видео дотроос биетийн дүрсийг олж, таних үйл явц юм. Хүн зураг дотроос хэд хэдэн дүрсийг өөр өөр өнцгүүдээс, олон янзын хэмжээтэй, эргэсэн тохиолдлуудад төвөггүй таньж чаддаг [1]. Компьютерийн хараанд (vision) эдгээр нь одоогоор хараахан бүрэн шийдэлд хүрээгүй бөгөөд хэд хэдэн аргуудаар шийдэхийг оролдсоор байна [2-4]. Тэдгээр аргуудаас доор дурдвал:

Дундаж квадрат хазайлт: Эрчмийн утгуудын ялгаврын квадратуудыгхооронд нь нэмж нийт пикселийн тоонд хувааснаар дүрснүүдийн ялгааг илтгэсэн коэффициент гарч ирэх ба 0-рүү дөхөх тусам хоёр дүрс хоорондоо төстэй байна.

$$DKX = \frac{1}{n} \sum_{i=1}^n (I_i - \bar{I})^2$$

Нормальчилсан Кросс Корреляци(НKK): Дүрс тус бүрийн цэг дээрх эрчмийн утгуудаас тэдгээрийн дунджийг хасаж хоорондох үржвэрийг олсны дараа, стандарт тархалтуудын үржвэрт хувааж, тэдгээрийн нийлбэрийг олсноор кросс корреляцийг олно. Утгарсан корреляцийн утгыг нийт пикселийн тоонд хувааж өгснөөр нормальчлах юм.

$$HKK = \frac{1}{n} \sum_{i=1}^n \frac{(I_i - \bar{I}) \cdot (J_i - \bar{J})}{\sigma_I \cdot \sigma_J}$$

Энд $\bar{I}$, $\bar{J}$ анхны дүрс, σ_I , σ_J харьцуулж буй дүрс байна.

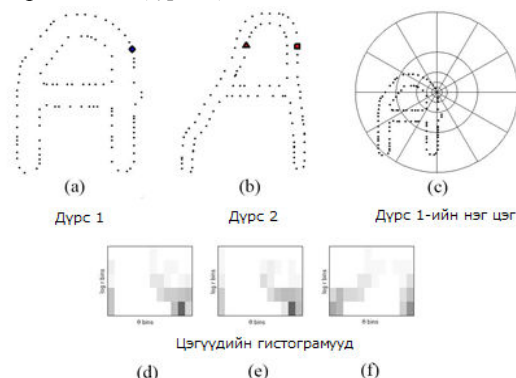
Хэнийэрчим(ХЭ): Эрчмийн утгуудыг—төрлийн функцээр хувиргасны дараа тэдгээрийн ялгаврын квадратуудыг олж нэмнэ. Энэ хэмжүүрийн давуу тал нь эрчмийнутгууд ойролцоо, олон харьцуулах хувилбартай үед нарийвчлал нь ихэсдэг.

$$HE = \frac{1}{n} \sum_{i=1}^n \frac{1}{1 + |f(I_i)|}$$

II. ХЭЛБЭР ТААРУУЛАХ

Хэлбэрийн хам орчныг тааруулахарга нь нэг дүрсэн дээрх тодорхойлогчуудыг бодоод өөр дүрс дээр эдгээр тодорхойлогчуудад таарах цэгүүд байгаа эсэхийг дараах 6 алхмаар шалгадаг. Үүнд [5-7]:

1. Дүрсний ирмэгийг ялган авч тэдгээрээс жигд тархсан ширхэг цэгийг ялган авах.
2. Цэг тус бүрийн хувьд бусад цэг рүү шулуун татаж тэдгээр цэгүүдийн хоорондох Евклид зай болон өнцөг —г бодож олсны дараа утга тус бүрийг —ийн дунджид хувааж нормальчилж, өнцгийг эерэг тэнхлэгтэй харьцуулж хэмжинэ.
3. векторыг логарифмд шилжүүлэн, логарифм туйлын координатын системд шилжүүлнэ.
4. Цэг тус бүрийг координатын эх гэж үзэн өгөгдсөн θ , тасалбар дахь цэгүүдийн тоог тодорхойлно. Цэг тус бүрийн хам орчин нь бусад 1 цэгүүд хүртэл координатын эхээс хэмжсэн логарифм туйлын гистограм байна(зураг 1).



Зураг 1. Хэлбэрийн хам орчны дүрслэлт [5]

5. Тус бүрийн дүрсэн дээр болон гэсэн хоёр цэгийг авч үзье. Эдгээр цэгүүд нь нормальчилсан тасалбартай θ , гэсэн гистограмуудтай. Хэлбэрийн хам орчин нь гистограм хэлбэрээр илэрхийлэгдэж байгаа учраас тест статистикийг хэрэглэнэ(томъёо 4).

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i}$$

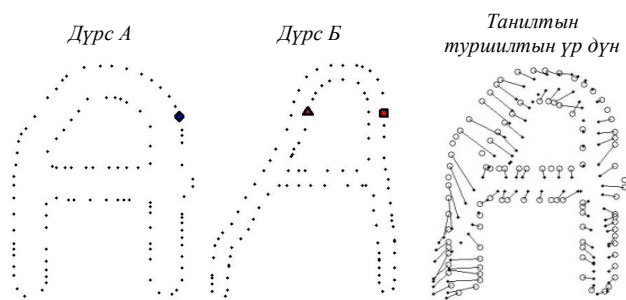
Энд O_i , E_i цэгүүдийг холбох зардал бөгөөд 0-ээс1-ийн хооронд байна. Үүнээс гадна харагдах байдал дээр тулгуурласан зардал байж болно. Жишээнь: Тангенс өнцгүүдийн ялгаа.

$$\chi^2 = \sum_{i=1}^n \frac{1}{2} \frac{\sin^2 \theta_i}{\cos^2 \theta_i} \quad 5$$

6. Дүрс тус бүрийн θ , цэгүүдийг хооронд нь нэг нэгэнд нь харгалзуулж нийт зардлыг хамгийн бага байхаар тааруулалт хэрэгтэй(томъёо 7).

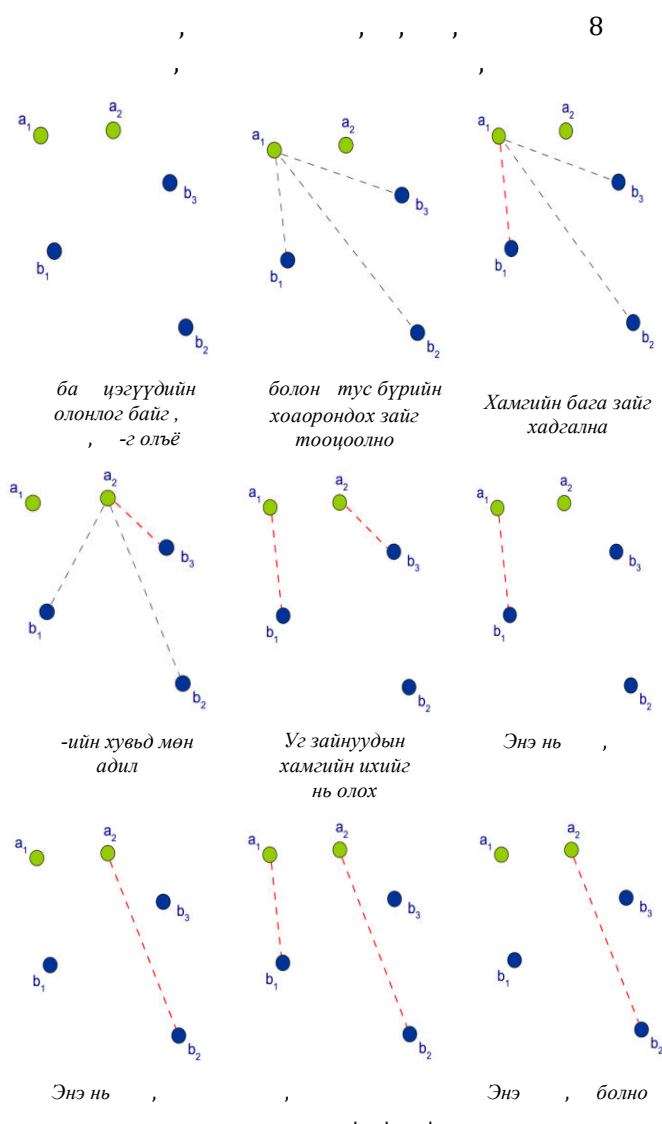
$$\chi^2 = \sum_{i=1}^n \frac{1}{2} \frac{\sin^2 \theta_i}{\cos^2 \theta_i} \quad 7$$

Бүх боломжит хувилбаруудыг туршиж үзэн хамгийн бага зардалтай цэгүүдийг хооронд нь харгалзуулсан –ийг бодох олон янзын алгоритмууд байдаг [8].



Зураг 2. Тааруулалтын үр дүн [8]

Хаусдорфф зайг тодорхойлох ерөнхий томъёо (8) хэлбэртэй байна [9]. Үүнд:



Зураг 3. Хаусдорфф зайг бодох зарчим [9]

III. ОНЦГОЙ ШИНЖ ТААРУУЛАХ

Дүрсэн дэх ямар ч дүрсний хувьд тохирох цэгүүдийг нь гарган авч дүрсний онцгой шинжийн тодорхойлогчийг гарган авч болдог. Уг гарган авсан тодорхойлогчийг ашиглан тухайн дүрсийг өөр олон

дүрстэй зургаас олоход ашигладаг [10]. Иймэрхүү цэгүүд нь ихэвчлэн дүрсний тод хэсгүүдэд, ирмэг, булан, нугалаанд байдаг. Онцгой шинжийг тааруулах нь онцгой шинж чанарыг тодорхойлогч болон тааруулагч гэсэн хоёр алгоритмаар дамжин гүйцэтгэгддэг [11].

SIFT (Scale Invariant Feature Transform) [10-11]: Хэмжээнээс үл хамаарсан шинж чанарын хувиргалт нь жигд хэмжээний өөрчлөлт, өнцгийн өөрчлөлтөд ажрахгүй ба аффейн өөрчлөлтөд бага зэрэг тэсвэртэй. Дараах шинжүүд түлхүүр үе шатанд хамаардаг. Үүнд:

Хэмжээний өөрчлөлтөд тэсвэртэй онцгой шинжийг илрүүлэх: Дүрснээс онцгой шинжийг ялган авах Ловын арга нь дүрсийг онцгой шинж бүхий векторуудын олонлог болгон хувиргах юм [10]. Эдгээр векторууд тус бүр нь хэмжээ, өнцөг, хөрвөлтийн өөрчлөлтөд тэсвэртэй ба гэрэлтүүлгийн өөрчлөлтөд хэсэгчлэн тэсвэртэй [12]. Эх дүрсийг хэд хэдэн удаа жижгэрүүлж тэдгээр дээр Гауссийн хувиргалт хийсний дараа хоорондын зөрүүний максимум ба минимум утгуудыг түлхүүр байршлууд гэж авч үздэг [10].

Онцгой шинжийг тааруулах ба индекслэх: Индекслэх үйл явцад SIFT түлхүүр цэгүүдийг хадгалж, харьцуулж буй дүрснээс олох үйлдлүүд орно [10]. Лов хэмжээст модны алгоритмын нэг хувилбар болох хамгийн сайн савны хайлтын аргыг энэ судалгаанд ашигласан. Энэ аргаар хамгийн ойрын хөршийг багахан тооцоо ашиглан өндөр магадлалтайгаар олдог. Нэр дэвшсэн хамгийн сайн тохироог өмнө нь гаргаж авсан түлхүүр цэгүүдтэй харьцуулан хамгийн ойрын хөршүүдийг олсноор тодорхойлно [12-13]. Хамгийн ойрын хөршүүдийг тухайн онцгой шинжийн вектороос Евклид зай нь хамгийн бага байх түлхүүр цэгүүд гэж авч үзнэ. Тухайн тохироо үнэн зөв байх магадлалыг хамгийн ойрын хөрш ба хоёр дахь ойрын хөршийн харьцаагаар тодорхойлж болно [14].

Хюү хувиргалтаар бөөгнөрөл олох: Хюү хувиргалт ашиглан найдаж болохуйц загварын цэгүүдийн бөөгнөрлийг олж тодорхой загварын поз байрлалыг заах түлхүүр цэгүүдийг хайдаг. Загварын поз гэдэг нь биетийн байршил болон чиглэлийн комбинацыг хэлнэ. Биетийн нэг позыг зааж буй хэд хэдэн түлхүүр цэгүүдийн бөөгнөрөл олдвол тухайн биетийг зөв тодорхойлох магадлал нь аль ч түлхүүр цэгийнхээс илүү байдаг [10-14].

IV. ШУГАМАН ХБК АРГААР ЗАГВАР ШАЛГАХ

Илэрсэн бөөгнөрөл болгоныг дүрсний загвартай нь холбох аффейн (affine) хувиргалтын параметруудын хувьд шугаман ХБК-ын аргаар шийдийг олж шалгана [15-16]. Загварын цэг 1-ийн дүрс дэх –д харгалзах аффейн хувиргалт нь 9 байдлаар тодорхойлогдоно.

$$\begin{pmatrix} 1 & 2 & 9 \\ 3 & 4 & 9 \end{pmatrix}$$

Энд загварын хөрвүүлэлт нь $[tx \ ty]^T$ байх ба аффейн эргэлт, хэмжээний өөрчлөлт, суналтыг $m1, m2, m3, m4$

параметруудээр илэрхийлсэн. Хувиргалтын параметруудийн голохын тулд (9) тэгшитгэлийг хувиргаж үл мэдэгдэгчүүдийг баганан вектор болгож 10 томъёогоор тооцоолно.

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ \vdots \\ \vdots \end{pmatrix} = 10$$

(10) томъёонд нэг цэг таарч байгаа боловч цааш хэдэн ч цэг нэмж болно. Нэмэгдсэн цэг бүрийн тоогоор эхний болон сүүлийн матрицад хоёр, хоёр мөр тус тус нэмэгдэнэ. Шийд гаргаж авахын тулд дор хаяж гурван цэг таарч байх шаардлагатай. Энэ шугаман системийг (11) байдлаар бичиж болно.

11

Энд A нь $m \times n$ матриц, b нь үл мэдэгдэх хэмжээт параметр вектор, x нь хэмжээт хэмжих вектор байна.

Иймд векторыг минимумчлах нь (12) нормаль тэгшитгэлийн шийд болно.

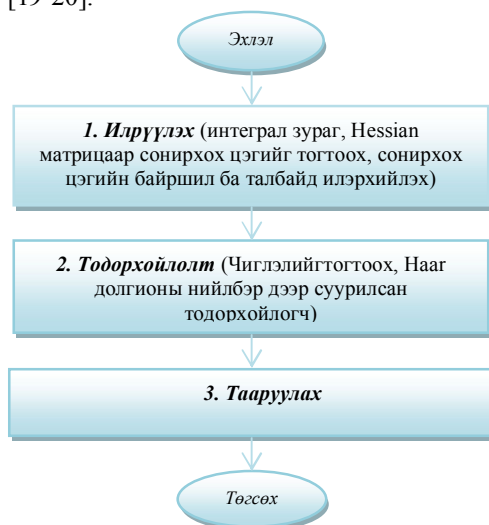
б 12

Шугаман тэгшитгэлийн системийн шийд нь (13) томъёогоор тооцоологдох ба энэ нь дүрсэн дээрх биет ба түүнд харгалзах загвар дээрх биет хоёрын хоорондох зайн квадратуудын нийлбэрийг минимумчилна.

б 13

Гадуурх цэгүүдийг илрүүлэх: Параметр шийдийг олсноор дүрсний болон загварын онцгой шинж бүрийн хоорондох харьцааг авч үзэн хий илрэлтүүд буюу биетээс гадуурх цэгүүдийг хасаж болно [17]. Гадуурх цэгүүдийг хасах болгонд шугаман ХБК-ын аргаар үлдсэн цэгүүд дээр дахин бодох замаар давтан гүйцэтгэнэ [18]. Хэрэв гадуурх цэгийг хассаны дараа гурваас бага цэг үлдэж байвал уг тохироог хүчингүйд тооцно [17].

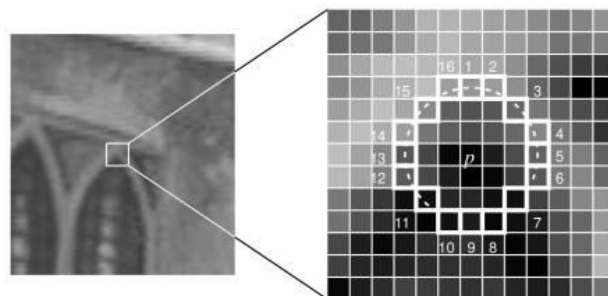
SURF (Speeded Up Robust Features): SIFT илрүүлэгчээс зарим нэг санааг авсан уг илрүүлэгчийн стандарт хэлбэр нь SIFT-ээс хэд дахин хурдан ба төрөл бүрийн дүрсний хувиргалтуудад тэсвэртэй гэж зохиогчид нь үздэг [19-20].



Зураг 4. SURF алгоритмын гол үе шат [19]

FAST (Features from Accelerated Segment Test): FAST нь ирмэг илрүүлэх арга бөгөөд онцгой цэгүүдийг ялган авч хадгалах үйлдлийг гүйцэтгэдэг [21]. FAST илрүүлэгчийн хамгийн чухал давуу тал нь тооцооллын хурд юм. Гауссийн хувиргалтын өөрчлөлтөөр онцгой цэг ялган авдаг SIFT, SUSAN, Harris зэрэг илрүүлэгчдээс хавьгүй хурдан ба машин суралцах алгоритмуудтай хавсаргавал илүү сайн үр дүн үзүүлдэг [21]. FAST ирмэг илрүүлэгч нь бага тооцоолох хүчин чадал шаарддаг учир бодит цагаар дүрс бичлэг боловсруулахад маш тохиромжтой. FAST алгоритмын булан илрүүлэх алхмууд [21]:

1. пикселийг сонирхох цэг мөн эсэхийг тогтоохоор сонгож авъя. Түүний эрчим нь байг.
2. Оновчтой гэсэн босго утга сонгон авъя.
3. Сонирхож буй пикселийг хүрээлэн буй 16 пикселийг сонгон авъя.
4. Хэрэв бүгд –ээс дээш эрчимтэй эсвэл – –ээс доош эрчимтэй тооны тасралтгүй цэгүүд тухайн пикселийг хүрээлэн байвал тухайн пикселийг булан гэж үзнэ. Зураг 5 дээр –ийг 12 байхаар сонгож авсан.
5. Булан биш пикселүүдийг ялган хасах өндөр хурдны тестийг явуулна. Уг тест нь 1, 9, 5, 13 гэсэн дөрвөн пикселийг шалгаж үздэг. Эхлээд 1, 9 хоёрыг шалгаж үзээд хэрэв тогтоосон босгыг давж байвал 5, 13 хоёрыг шалгадаг. нь булан байхын тулд дээрх дөрвөн пикселийн ядаж гурав нь –ээс тод эсвэл – –ээс бүдэг байх шаардлагатай. Хэрэв тийм биш бол булан байж таарахгүй.



Зураг 5. FAST алгоритмын булан илрүүлэх үйл явц [21]

MSER (Maximally Stable Extremal Regions) [22]: MSER нь зураг дахь төвийлтийг илрүлдэг алгоритм юм. Хоёр өөр өнцгөөс авсан нэг ижил дүрсний адил төстэй, харгалзах цэгүүдийг олох зорилготой. MSER алгоритмын гол үйлдэл нь дүрснээс хамгийн том үргэлж хэсгүүдийг олох явдал юм. Дүрсийг олон жижиг хэсэг болгон хуваасны дараагаар нэг нэгээр нь 0 эсвэл максимум хүртэл босгоор шалган үзэж жижиг хэсгүүдийг нийлүүлэн том хэсэг болгоод хэрэв тогтвортой байвал хадгалдаг. Талбайн өсөх хурд нь 0 рүү тэмүүлж зогсонги байдалд орох үед тэнд талбайн ирмэг байна гэж тогтооно. Эх дүрс болон харьцуулж буй дүрснүүд дээр эдгээрталбайнуудыг тохирч буй эсэхийг шалгана.

ORB (Oriented FAST and Rotated BRIEF) [23]: ORB нь ерөнхийдөө FAST түлхүүр цэг олох алгоритм ба BRIEF тодорхойлогч алгоритмуудыг өөрчилж гүйцэтгэлийг нь сайжруулсан нийлмэл алгоритм юм. FAST алгоритм нь чиглэл тодорхойлогчийг учир

эргэлтэнд тэсвэргүй энэ сул талыг засахын тулд доорхи хоёр аргыг ашигладаг. Үүнд:

Эрчмийн центройд[23]: Булангийн эрчим нь төвийн эрчмээсээ тодорхой хэмжээний зөрүүтэй гэж үзэн центройдоос булан хүртэл вектор татах ба энэ векторыг ашиглан чигийг тооцоолно. Ингэхийн тулд x болон y тэнхлэгийн дагуух моментуудыг олох шаардлагатай:

$$M_x = \sum_{i=1}^N x_i \cdot I(x_i, y_i) \quad (14)$$

Энэ моментуудыг ашиглан центройдг олвол:

$$x_c = \frac{M_x}{M_0}, \quad y_c = \frac{M_y}{M_0} \quad (15)$$

Булангийн төв O -оос центройд S хүртэл вектор татвал тухайн хэсгийн чиглэл нь:

$$\theta = \arctan\left(\frac{y_c - y_0}{x_c - x_0}\right) \quad (16)$$

BRIEF (Binary Robust Independent Elementary Features)[24]: BRIEF тодорхойлогч нь зураг хэдхэн градус эргэхэд л тохироог зөв олж чадахаа байдаг тул энэ сул талыг засварлах шаардлагатай. Үүний тулд түлхүүр цэгүүдийн чиглэлийн дагуу BRIEF-ийг эргүүлэх шаардлагатай юм. , байршил дээр хийсэн ширхэг бинар тестийн үр дүнд олдсон түлхүүр цэгүүдийн олонлогийн хувьд x матриц тодорхойлно.

$$R_{\theta} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (17)$$

Хэсгийн чиглэл болох θ -аар харгалзах эргүүлэгч матриц R_{θ} -ийг ашиглан “залсан” матриц болох R -г 18 томъёогоор гаргаж авна.

$$R_{\theta} = R_{\theta_1} \cdot R_{\theta_2} \cdot \dots \cdot R_{\theta_{18}} \quad (18)$$

Одоо BRIEF оператор нь 19 хэлбэртэй болно.

$$BRIEF(x, y) = \sum_{i=1}^{19} (I(x_i, y_i) - I(x_j, y_j)) \cdot w_i \quad (19)$$

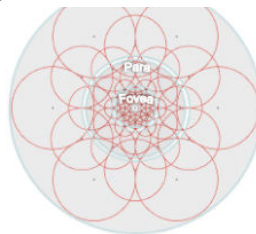
Өнцгийг дискрет хэлбэрт шилжүүлж $\frac{\pi}{12}$ (12 градус) инкрементээр эргүүлсэн BRIEF дүрсүүдийг агуулсан таблиц бэлдэнэ. Түлхүүр цэгүүдийн чиглэл θ -аарч л байвал тэдгээрийн тодорхойлогчийг бодох зөв хэрэглэгдэнэ.

BRISK (Binary Robust Invariant Scalable Keypoints): ORB-оос илүү найдвартай SURF-аас илүү хурдтай алгоритм гаргаж авах зорилгоор уг алгоритмыг дэвшүүлсэн [25]. BRISK нь түлхүүр цэг илрүүлэгч бөгөөд тодорхойлогч биш юм. BRISK илрүүлэгчийг дурын тодорхойлогчтой хавсран хэрэглэх боломжтой ба энэ нь тухайн шийдэх гэж буй асуудалд тохирсон комбинацыг гаргаж авах боломжийг бүрдүүлж байгаа юм. Түлхүүр үе шатанд гурван шат орно [25]. Үүнд:

1. Scale огторгуйд түлхүүр цэгүүдийг илрүүлэх
2. Түлхүүр цэгүүдийг тодорхойлох
3. Түлхүүр цэгүүдийг тааруулах

FREAK (Fast Retina Keypoint): FREAK ялгах загвар нь гараар бүтээгдсэнээрээ BRISK-тэй төстэй ба машин сурах аргууд ашиглагдаараа ORB-тоймөн төстэй. FREAK нь мөн чиг тогтоохдоо BRISK-тэй төстэй процесс ашигладаг [26].

Нүдний шинжих загвар: Пикселийн эрчмийг шинжих олон янзын загварууд бий. BRISK нь дугуй хэлбэрийн загвар ашигладаг бол FREAK нь хүний нүд шиг шинжих загварыг ашигладаг [25-26]. Мөн л дугуй хэлбэрийн шинжих загвар боловч төврүүгээ ойртох тусам цэгүүдийн нягт ихэсдэг. Гадна тойрогтоо анхан шатны шинжилт хийж тодорхой босгыг давбал доторх тойргуудаар нарийн шинжилгээ хийдэг нь хүний нүдний ажиллагаатай төстэй (зураг 6).

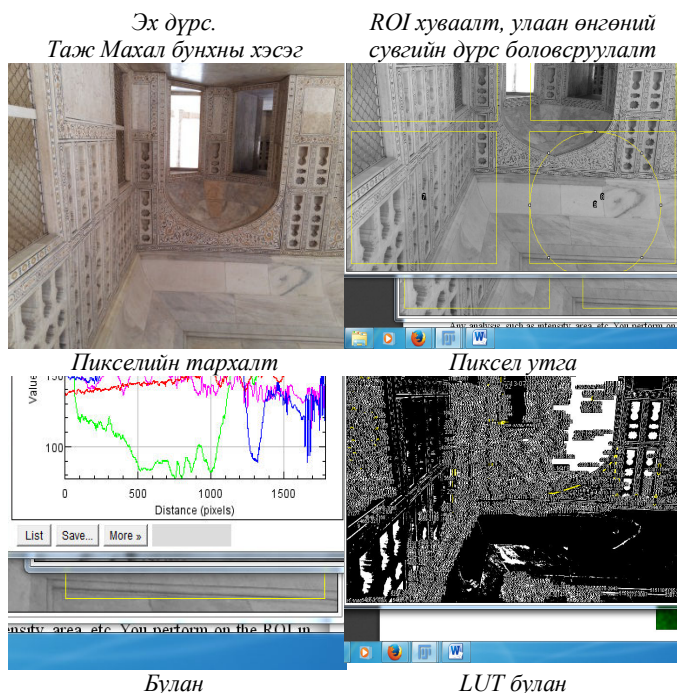


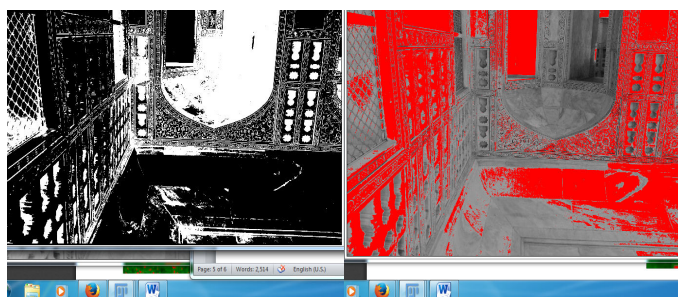
Зураг 6. FREAK алгоритмын шинжих загвар [26]

FREAK алгоритмын цагираг бүр нь логарифм туйлын координатын систем юм. Уг алгоритмоор дүрсийг шинжлэхдээ [26]:

1. Гауссийн оператороор эрчмийг тооцоолно.
2. Илэрсэн цэгүүдийг төв рүү ойртуулж илүү нарийн тойргуудаар шинжилнэ.
3. Эцсийн төвд хүрсэн цэгүүдийг оновчтой гэж үзэн хадгална.
4. Цэгийн байршилыг бинар цуваанд хөрвүүлж хадгална.
5. Эх дүрс ба харьцуулж буй дүрсний бинар цуваан дээр Hamming зайг бодно.
6. Эргэлтийн алдааг урдчилан 45° байдлаар эргүүлсэн дүрсүүдтэй харьцуулж арилгана.
7. Хамгийн бага Hamming зайтай хоёр цэгийг оновчтой танилт гэж үзнэ.

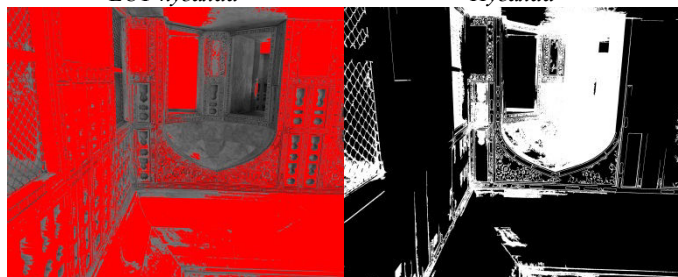
FREAK нь харьцангуй сүүлийн үеийн алгоритм бөгөөд оновчлолын туршилтын үр дүн ховорбайдаг. Судалгааны туршилтын дүрсний зарим хэсгийг 7-р зурагт үзүүлэв.





LUT нугалаа

Нугалаа



Mask

Дугуй нүх



Count max

Ирмэг



Зураг 7. Ижил төстэй дүрснүүдийг таних туршилтын дүрсний хэсэг

Дүгнэлт

Тус судалгааны ажилд дүрсний ижил төстэй хэмжигдэхүүнийг тодорхойлох аргачлалуудын талаар өгүүлсэн. Мөн дүрс боловсруулалтын ирмэг илрүүлэлт, шүүлтүүр, танилтын аргуудын харьцуулсан судалгаа хийж туршилтын үр дүнгүүдийг гарган авсан [27]. Дүрс танилтын SIFT, SURF, ORB, FAST алгоритмууд ба тэдгээрийн хосмогуудыг туршиж үзсэний үндсэн дээр ORB ба SURF алгоритмуудын хосмог нь хамгийн түргэн хугацаанд оновчтой үр дүн үзүүлж байсан.

Ашигласан бүтээл

- [1] Reinhard Klette (2014), "Concise Computer Vision", Springer, ISBN 978-1-4471-6320-6
- [2] Linda G. Shapiro and George C. Stockman (2001), "Computer Vision", Prentice Hall, ISBN 0-13-030796-3
- [3] Tim Morris (2004), "Computer Vision and Image Processing", Palgrave Macmillan, ISBN 0-333-99451-5

- [4] Bernd Jähne and Horst Haußecker (2000), "Computer Vision and Applications", A Guide for Students and Practitioners, Academic Press, ISBN 0-13-085198-1
- [5] S. Belongie and J. Malik (2000), "Matching with Shape Contexts" (PDF), IEEE Workshop on Contentbased Access of Image and Video Libraries (CBAIVL-2000)
- [6] S. Belongie, J. Malik, and J. Puzicha (April 2002), "Shape Matching and Object Recognition Using Shape Contexts" (PDF), IEEE Transactions on Pattern Analysis and Machine Intelligence (24): 509–521, doi:10.1109/34.993558
- [7] H. Chui and A. Rangarajan, (June 2000), "A new algorithm for non-rigid point matching", CVPR, pp 44–51
- [8] R. Jonker and A. Volgenant, (1987), "A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems", Computing 38 (4): pp 325–340, doi:10.1007/BF02278710
- [9] M.J.D. Powell, (1995), "A Thin Plate Spline Method for Mapping Curves into Curves in Two Dimensions", Computational Techniques and Applications (CTAC '95)
- [10] Lowe, David G, (1999), "Object recognition from local scale-invariant features", Proceedings of the International Conference on Computer Vision, pp 1150–1157, doi:10.1109/ICCV.1999.790410
- [11] U.S. Patent 6,711,293, "Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image", David Lowe's patent for the SIFT algorithm, March 23, 2004
- [12] Lowe, David G, (2004), "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision 60 (2): pp 91–110, doi:10.1023/B:VISI.0000029664.99615.94
- [13] Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., Poggio, T., "A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex", Computer Science and Artificial Intelligence Laboratory Technical Report, December 19, 2005 MIT-CSAIL-TR-2005-082.
- [14] Beis, J.; Lowe, David G, (1997), "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces" (PDF). Conference on Computer Vision and Pattern Recognition, Puerto Rico: sn. pp. 1000–1006. doi:10.1109/CVPR.1997.609451.
- [15] Goldberger, Arthur S, (1964), "Classical Linear Regression", Econometric Theory, New York: John Wiley & Sons, pp156–212, ISBN 0-471-31101-4.
- [16] Lawson, C. L.; Hanson, R. J, (1974), "Solving Least Squares Problems", Englewood Cliffs, NJ: Prentice-Hall, ISBN 0-13-822585-0.
- [17] Lowe, D.G., "Local feature view clustering for 3D object recognition", IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, 2001, pp 682–688
- [18] Г.Ганчимэг, "Дүрсний ирмэг илрүүлэх шугаман алгоритм ба параметрийн загвар", ММТ 2015, х 127-131
- [19] US 2009238460, Ryuji Funayama, Hiromichi Yanagihara, Luc Van Gool, Tinne Tuytelaars, Herbert Bay, "Robust interest point detector and descriptor", published 2009-09-24
- [20] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded Up Robust Features", ETH Zurich, Katholieke Universiteit Leuven
- [21] Edward Rosten and Tom Drummond, "Machine learning for high speed corner detection", 2006
- [22] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions", 2002
- [23] Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski, "ORB: an efficient alternative to SIFT or SURF", 2011

- [24] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua, "*BRIEF: Binary Robust Independent Elementary Features*", 11th European Conference on Computer Vision (ECCV), Heraklion, Crete, LNCS Springer, September 2010
- [25] Leutenegger Stefan, Margarita Chli, and Roland Y. Siegwart, "*BRISK: Binary robust invariant scalable keypoints*", Computer Vision (ICCV), 2011
- [26] Stroustrup, Bjarne, "*C++ Applications*", 2014
- [27] G.Ganchimeg, "*History Document Image Background Noise and Removal Methods*", Journal of International Journal of Knowledge Content Development & Technology, Vol.5, №2, 2015, pp 11-24