

Построение модели прибрежного рельефа в диапазоне наблюдаемых колебаний уровня воды по контурам береговых линий, выделяемых на спутниковых изображениях среднего разрешения

Алексей Евгеньевич Хмельнов

Институт динамики систем и теории управления имени В.М. Матросова
Сибирского отделения
Российской академии наук
(ИДСТУ СО РАН)
<http://idstu.irk.ru>

13 ноября 2024

Задача построения 3D моделей прибрежного рельефа

Требуется объединить информацию о наземном и подводном рельефе.

Много источников информации о наземном рельефе, но очень мало о подводном.

Модель прибрежного рельефа

Использование:

- Оценка последствий изменения уровня воды на прибрежные и подводные объекты (водозаборы, пристани, строения);
- Моделирование процессов затопления/обмеления.

Для таких проектов требуется наибольшая точность именно этой части модели рельефа.

Источники информации о подводной части рельефа:

- лоции, карты – неточно, недостаточный шаг изобат (например, на лоции 2 м, 5 м и всё);
- многолучевой эхолот – для него мелко;
- эхолот и т.п. – требует посещения каждой измеряемой точки, долго;
- анализ спутниковых снимков для определения глубин по цвету воды и т.п. – м.б. нужны снимки высокого разрешения, требуется подбор большого числа параметров;
- контура береговых линий при различных уровнях воды;
- топобатиметрический лидар – око видит, да зуб неймёт.

Данные ДЗЗ среднего разрешения для исследования прибрежного рельефа

- Позволяют получить наиболее широкий охват территории.
- Съёмка такими спутниками, как Sentinel 2 и LandSat выполняется *регулярно*, и такие данные распространяются *свободно* при разрешении 10 м/пиксел для основных каналов.
- Может показаться, что такого разрешения не хватит, для решения рассматриваемой задачи. Покажем, что **это не так**.

Контура береговых линии

Могут рассматриваться, как изолинии прибрежного рельефа.

при следующих условиях:

- уровень воды должен быть постоянен вдоль береговой линии в каждый момент времени – озеро, водохранилище. Иначе, для правильной интерпретации контуров береговых линий рек требуется модель течения (такое мы пока не делали);
- волнением можно пренебречь;
- приливы и сейши малы (так, для Байкала приливы до 32 мм и сейши до 60 мм – существенно меньше волнения);
- отсутствие льда и снега (снимки за позднюю осень, зиму, и почти всю весну не подходят, а как раз к концу весны уровни водохранилищ снижаются максимально);
- наличие колебаний уровня воды с течением времени (чем больше - тем лучше) – получаем несколько изолиний для разных уровней;
- наличие гидропостов и информации от них.

Обнаружение береговых линий

Сначала рассмотрим ранее полученные результаты.

Традиционный подход к выделению береговых линий

Шаги обработки снимка

- 1 Вычислить растр NDWI (Normalized Difference Water Index) (для Sentinel-2 используем каналы $B03$ – *Green* и $B08$ – *Visible and Near Infrared*: $NDWI = \frac{B03 - B08}{B03 + B08}$);
- 2 Выполнить бинаризацию растра NDWI по порогу;
- 3 Векторизовать бинаризованный растр – получаем ступенчатые линии (┘);
- 4 Сгладить линии по некоторому порогу (↘).

После анализа полученных таким образом контуров был сделан вывод о **недостаточной точности растров с разрешением 10 м/пиксель** для решения задачи отслеживания изменений береговых линий.

Основная проблема

Изменение яркости любого пикселя **не влияет** на положение контура, пока это значение не перейдёт порог.

Пример результатов



Наш подход: выделение изолиний без бинаризации

Шаги обработки снимка

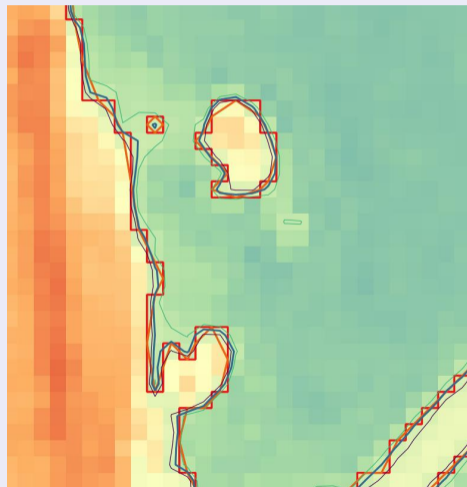
- 1 Вычислить растр NDWI (Normalized Difference Water Index) (для Sentinel-2 используем каналы *B03 – Green* и *B08 – Visible and Near Infrared*: $NDWI = \frac{B03 - B08}{B03 + B08}$);
- 2 Построить изолинии растра NDWI с шагом 0.05 от 0.0 до 0.25; (`gdal_contour`) (/);
- 3 Выбрать наилучший порог (обычно 0.1).

В результате получаем субпиксельный уровень точности, превышающий 10 м.

Основное преимущество

Любое **изменение яркости** пикселя на границе областей **влияет** на положение контура.

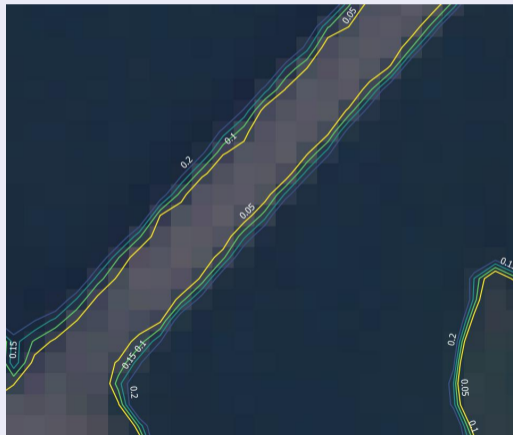
Сравнение результатов



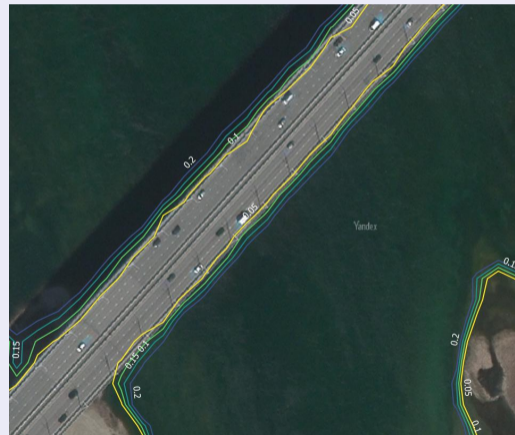
Изолинии растра NDWI в районе моста

Наиболее подходящий порог можно выбрать по объектам, имеющим чёткую границу с водой при любом направлении съёмки:

на фоне ТСИ



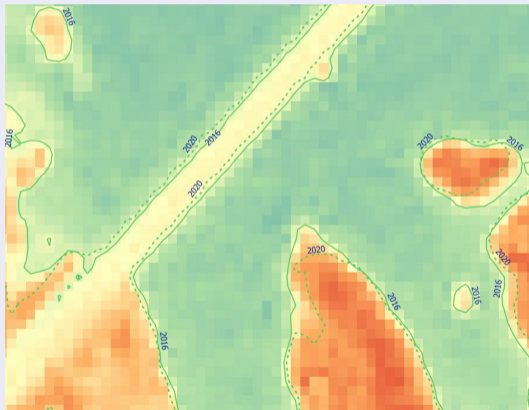
на фоне Яндекс Спутник



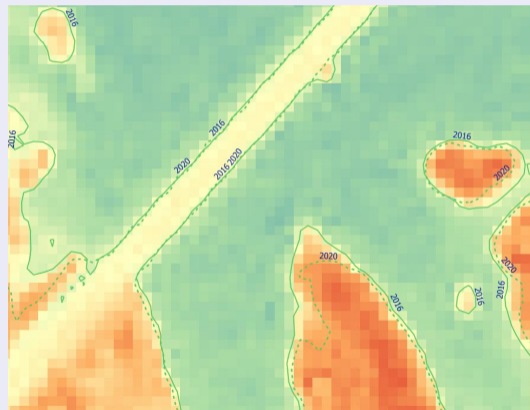
Точность горизонтального позиционирования

Её оказалось недостаточно для работы на субпиксельном уровне. Пример: 2016 г. – сплошная линия, уровень воды ниже, 2020 г. – пунктир, уровень воды выше.

Исходное положение



Пунктир сдвинут вручную



С учётом разных возможных направлений съёмки со спутника на изображениях нет надёжных ориентиров, позволяющих найти сдвиг.

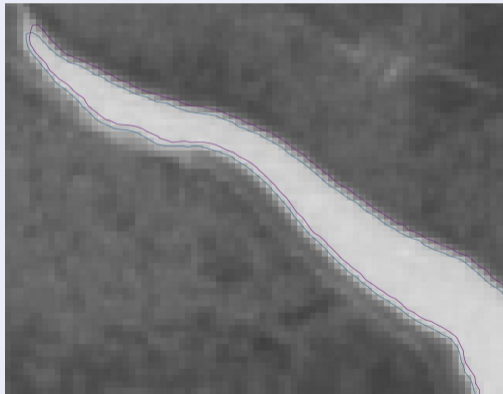
Векторный алгоритм поиска наилучшего сдвига

- 1: Выбрать объект F_t целевой карты, имеющий наибольшее пересечение с исходной
- 2: Выбрать наибольший объект F_s исходной карты среди достаточно близких к F_t
- 3: $Shift \leftarrow 0$
- 4: **for** $step \in 1..MaxSteps$ **do**
- 5: $N \leftarrow 0$; $Sum_{\Delta} \leftarrow 0$
- 6: **for all** $P_s \in F_s$ **do**
- 7: $P_t \leftarrow ClosestPoint(F_t, Shift + P_s)$
- 8: **if** $Distance(P_t, P_s) \leq D_{max}$ **then**
- 9: $N \leftarrow N + 1$
- 10: $Sum_{\Delta} \leftarrow Sum_{\Delta} + P_t - P_s$
- 11: **end if**
- 12: **end for**
- 13: $\Delta \leftarrow Sum_{\Delta} / N$
- 14: $Shift \leftarrow Shift + rate * \Delta$
- 15: **if** $|\Delta| < \Delta_{max}$ **then**
- 16: **break**
- 17: **end if**
- 18: **end for**

Пример работы алгоритма

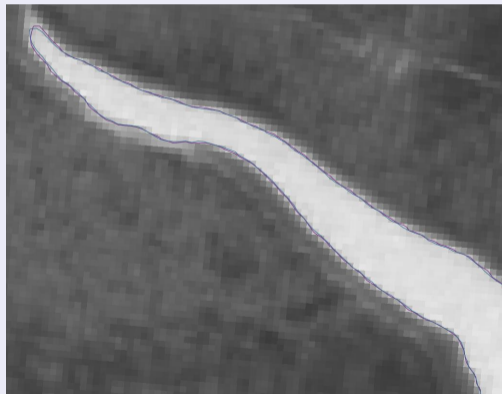
Использованы пересекающиеся изображения в разных проекциях (WGS 84/UTM zone 47N и WGS 84/UTM zone 48N) для разных моментов времени: 20180827 (уровень 396.41 м, синяя линия) и 20180825 (уровень 396.32 м, фиолетовая), изолинии на уровне 0.1, сдвиг (3.63 м, 8.86 м) найден за 15 шагов.

Исходное положение



на фоне NDWI для 20180827

Результат сдвига



на фоне NDWI для 20180825

Исходные данные

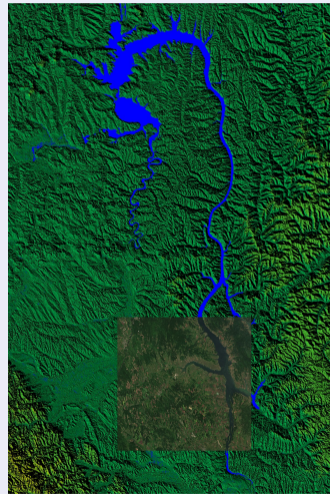
Далее рассмотрим результаты, полученные в 2024 г.

Объект исследования - Братское водохранилище

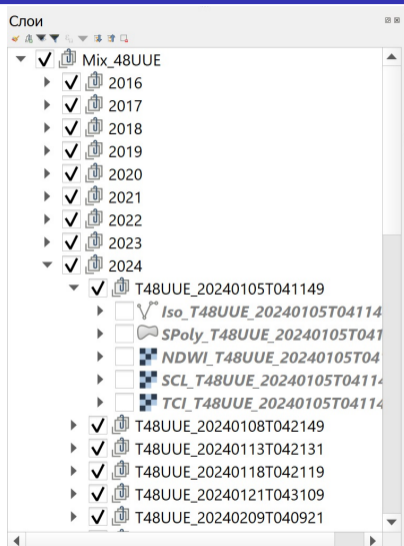
Диапазон колебаний уровня Братского вдхр. (ТО)

Год	min	Дата min	max	Дата max
2013	398.42	31.12.2013	399.54	12.09.2013
2014	396.62	31.12.2014	398.4	01.01.2014
2015	395.84	12.04.2015	396.93	05.09.2015
2016	395.58	02.04.2016	397.75	11.09.2016
2017	395.05	04.07.2017	396.25	04.01.2017
2018	394.52	10.04.2018	396.65	12.10.2018
2019	395.89	11.03.2019	400.02	11.10.2019
2020	397.32	28.04.2020	400.6	05.11.2020
2021	398.3	25.03.2021	402.03	31.08.2021
2022	399.32	07.04.2022	401.28	29.09.2022
2023	398.02	29.04.2023	402.01	12.10.2023
2024	398.89	02.04.2024	401.33	29.11.2024

Copernicus DEM, 48UUE



Исходные данные по трапеции 48UUE



Всего 461 снимок с облачностью менее 30%, 428 обработалось до L2A.

Использование L2A вместо L1C

Уровень обработки L1C



$$\vec{\sigma} = (13.2, 8.4, 5.8)$$

Разные лучшие уровни NDWI для разных снимков (0.1, 0.15, 0.05, 0.00).

Уровень обработки L2A



$$\vec{\sigma} = (23.9, 17.0, 12.2)$$

Лучший уровень NDWI $\equiv 0$.

Взаимная регистрация растров Sentinel с субпиксельной точностью

Задача поиска наилучшего сдвига относится к задачам взаимной регистрации снимков.

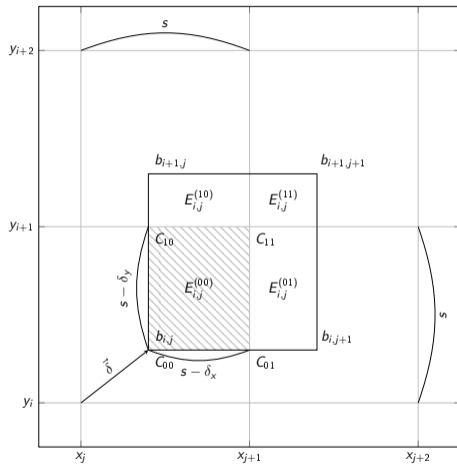
Для массовой обработки нескольких сотен снимков требуется максимально автоматизировать работу с ними, включая задачу взаимной регистрации снимков.

Выбираем эталонный качественный безоблачный снимок, покрывающий всю трапецию, для остальных находим субпиксельный сдвиг, минимизирующий отклонение между этими изображениями (сравниваем растры NDWI).

Растровый алгоритм поиска наилучшего сдвига в SubPxAlign

- 1 Используем билинейную интерполяцию для сопоставления функций изображениям.
- 2 Для 4-х возможных направлений сдвига $\vec{\delta}$ (4-х квадрантов с учётом знаков по δ_x и δ_y) в символьном виде могут быть найдены выражения для функции ошибок $E(\vec{\delta})$ – интеграла квадрата разности кусочно-билинейных функций. Это – полиномы до 6-й степени от x и y (при поиске и β, κ). Вычисляются по необходимости (при попадании а квадрант).
- 3 По этим выражениям в символьном виде находим $\nabla E(\vec{\delta})$.
- 4 Используем метод градиентного спуска для поиска минимума $E(\vec{\delta})$. При переходе между квадрантами вычисляем, если требуется, соответствующее выражение и переключаемся на него.
- 5 Останавливаемся при $|\nabla E(\vec{\delta})| < \epsilon$ или достижении макс. числа шагов.

Дополнительно есть версия алгоритма с компенсацией различий в освещённости, где минимизируется $E(\vec{\delta}, \beta, \kappa)$. Здесь β, κ - параметры линейного преобразования яркости одного из сопоставляемых изображений: $b'_{i,j} = \beta + \kappa * b_{i,j}$.



Выражения, найденные SubPxAIgen для T48UUE_20240708T040551

Сохранённые выражения (могут использоваться повторно):

[Stat]

N=120560400

StdDev=0.271677229140943

Avg=-0.582235551398922

[Common]

K2=739283.38846551+2.02679075300694E-7 x+3.42610292136669E-7 y-3.55415977537632E-7 xy+8.34211277833674E-9 x²+8.31437318993267E-9 y²+4.12683220929466E-10 x²y+5.11363396071829E-10 xy²+7.00936197972624E-10 x²y²-5.00222085975111E-12 y³-5.21822585142218E-11 xy³-2.8421709430404E-14 x³y²+7.67954588809516E-11 x²y³

BK=-3547778.11140287+5.355104804039E-8 x+2.37487256526947E-8 y-2.1420419216156E-8 xy

[QBr_PP]

C=2116369.70940294+153.825621662196 x-1.68224826967344 y-0.269937043543905 xy-2.00181744682095 x²-0.955998971988038 y²-0.0185591053859753 x²y-0.235138087444284 xy²-0.234373223700118 x²y²-0.600977290072251 x³-0.213211352764802 y³+0.0148746159923121 x³y+0.0417452624478756 xy³+0.0248630228566071 x³y²+0.0585214389278121 x²y³-0.0140638672273781 x³y³

K=-2293342.56537409-787.410250001587 x+31.8354816874489 y-123.718678183854 xy+1076.98427495172 x²+971.089128194488 y²-14.1277289391873 x²y+166.93477479044 xy²-259.802278012192 x²y²-72.9974141541016 x³-185.051087245929 y³+18.3593959497109 x³y-36.714620963039 xy³+26.9553198840642 x³y²+71.7230446865926 x²y³-10.5358951465273 x³y³

B=6365324.24633318+329.727485254407 x+178.580554038286 y+3.56609981134534 xy-7.18319184100255 x²-1.83293555909768 y²-1.15766289830208 x²y-0.653497886843979 xy²-0.130291500594467 x²y²

Area=5348733

[QBr_NP]

C=2116371.24524651+153.283116967417 x-1.48545875167474 y+1.31264934595674 xy-1.54820500330538 x²+0.0456417236130164 y²-0.891105568790181 x²y-0.727847846824943 xy²+0.439932930815075 x²y²-0.594194679965995 x³-0.0960242982341697 y³-0.0216604248596468 x³y-0.0422969058138278 xy³+0.00809840331334044 x³y²-0.0024397988171927 x²y³+0.00677941140315852 x³y³

K=-2292587.90505591-548.617014480289 x-1357.0589801399 y-301.823036057875 xy+897.635076959834 x²+418.301519146808 y²+305.846754925621 x²y+10.9850414394818 xy²-59.5181869355456 x²y²-69.4485426644302 x³+184.097141226836 y³-20.4206362853619 x³y+52.0447583487139 xy³+11.8352621527204 x³y²-66.9793699980364 x²y³+5.03650264323498 x³y³

B=6365145.89793244+326.518843900412 x+178.116247456521 y+2.8511828891933 xy-5.74955316772684 x²+0.232153290882707 y²-1.70961444824934 x²y+0.357458461076021 xy²+0.275975774973631 x²y²

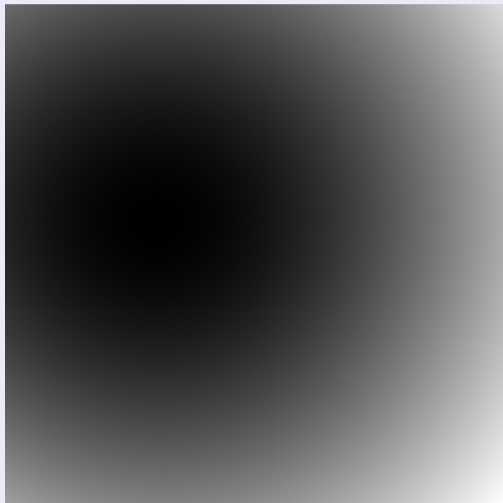
Area=5348733

Вывод SubPAlign для T48UUE_20240708T040551

```
Processing images 10980x10980
eta: 0.03, eps: 0.005, ScaleK: 8, ScaleB: 6
Avg: -0.582235551, Sigma: 0.271677229
1: K=(0, 0.105129669), S=(0, 0), G=(-0.00217440659, 0.0010348324) B=(B:0, K:1), BG=(B:7, 1.3696617, K:-27.9623868), eta=0.03
2: K=(0, 0.10011082), S=(0.0006652321978, -3.10449721E-5), G=(-0.0221166746, 0.00100779489) B=(B:-0.00594774181, K:1.01310737), BG=(B:6.85801554, K:-26.5731518), eta=0.033
3: K=(0, 0.0950677213), S=(0.00138217224, -6.43022035E-5), G=(-0.02250197177, 0.000978704142) B=(B:-0.012239861, K:1.02680915), BG=(B:6.56453609, K:-25.1124758), eta=0.0363
4: K=(0, 0.0900462828), S=(0.00219899403, -9.98291638E-5), G=(-0.0228797432, 0.000947466571) B=(B:-0.0185532266, K:1.04105263), BG=(B:6.25717269, K:-23.5838029), eta=0.03993
5: K=(0, 0.085096333), S=(0.00311328849, -0.000137661504), G=(-0.023299811, 0.00091399568) B=(B:-0.025793474, K:1.05576761), BG=(B:5.93688188, K:-21.9921682), eta=0.04329
6: K=(0, 0.080270367), S=(0.00413268608, -0.000177806936), G=(-0.0237051393, 0.000872124385) B=(B:-0.0330369646, K:1.07085987), BG=(B:5.60496751, K:-20.344379), eta=0.0483153
7: K=(0, 0.0756218871), S=(0.005286007, -0.000220238128), G=(-0.024108639, 0.000840057476) B=(B:-0.0405593448, K:1.08621838), BG=(B:5.26311147, K:-18.6491414), eta=0.05314683
8: K=(0, 0.0712033577), S=(0.00656330474, -0.00026488452), G=(-0.0245046957, 0.000799474126) B=(B:-0.0483292807, K:1.10170510), BG=(B:4.91339633, K:-16.9173266), eta=0.058461513
9: K=(0, 0.0670638409), S=(0.00799588633, -0.000311622978), G=(-0.0248868516, 0.00075430329) B=(B:-0.0563082965, K:1.11715833), BG=(B:4.55531481, K:-15.1617152), eta=0.0643076643
10: K=(0, 0.0632464336), S=(0.00958630163, -0.000360267254), G=(-0.0252478364, 0.000710911106) B=(B:-0.06445459237, K:1.13239293), BG=(B:4.20076576, K:-13.3972877), eta=0.0707384307
11: K=(0, 0.0597856846), S=(0.011382294, -0.00041055599), G=(-0.0255796424, 0.000662922277) B=(B:-0.0727052453, K:1.14720079), BG=(B:3.84402525, K:-11.6409194), eta=0.0778122738
12: K=(0, 0.056705227), S=(0.0133727041, -0.00046213948), G=(-0.0258736554, 0.000612491555) B=(B:-0.0810139216, K:1.16135402), BG=(B:3.49169505, K:-9.911565), eta=0.0858935012
13: K=(0, 0.0540158951), S=(0.0155873208, -0.000514564777), G=(-0.0261208466, 0.000559668705) B=(B:-0.0893157662, K:1.17460919), BG=(B:3.14762073, K:-8.22780122), eta=0.0941528513
14: K=(0, 0.0517146065), S=(0.018046673, -0.000567259181), G=(-0.0263120288, 0.000504524483) B=(B:-0.097547918, K:1.18671342), BG=(B:2.81577814, K:-6.61133928), eta=0.103568136
15: K=(0, 0.0497842519), S=(0.0207717608, -0.000619511841), G=(-0.0264381758, 0.00044714814) B=(B:-0.10564861, K:1.19741223), BG=(B:2.50012802, K:-5.08220519), eta=0.11392495
16: K=(0, 0.048194749), S=(0.0237837287, -0.000670543171), G=(-0.0264907946, 0.000387643326) B=(B:-0.11356047, K:1.20645895), BG=(B:2.20444171, K:-3.65990176), eta=0.125317445
17: K=(0, 0.0469052755), S=(0.0271034874, -0.000719031642), G=(-0.026462334, 0.00032612243) B=(B:-0.12123422, K:1.21362535), BG=(B:1.93210518, K:-2.36201003), eta=0.13784949
18: K=(0, 0.0458675206), S=(0.0307512987, -0.000763987355), G=(-0.0263466, 0.000262699625) B=(B:-0.128632531, K:1.2187287), BG=(B:1.68591575, K:-1.20315101), eta=0.151634109
19: K=(0, 0.045029606), S=(0.0347463419, -0.000803821578), G=(-0.0261391437, 0.000219748321) B=(B:-0.13573707, K:1.22156347), BG=(B:1.46788002, K:-0.193985628), eta=0.166797519
20: K=(0, 0.0443041815), S=(0.0391062862, -0.000836761288), G=(-0.0258375772, 0.000130568268) B=(B:-0.142534783, K:1.22206904), BG=(B:1.27905053, K:-0.659638505), eta=0.183477271
21: K=(0, 0.0437521297), S=(0.0438468943, -0.000860717598), G=(-0.0254417747, 6.20310334E-5) B=(B:-0.149053581, K:1.22017797), BG=(B:1.11940410, K:1.35727265), eta=0.201824998
22: K=(0, 0.0432253686), S=(0.0489816805, -0.000873237011), G=(-0.0249593196, -8.07334627E-6) B=(B:-0.155329262, K:1.21589779), BG=(B:0.987823953, K:1.90406756), eta=0.222007498
23: K=(0, 0.042728414), S=(0.0545216377, -0.000871444667), G=(-0.024378375, -7.97078969E-5) B=(B:-0.161242049, K:1.20929283), BG=(B:0.882115155, K:2.31098426), eta=0.244208248
24: K=(0, 0.0422386277), S=(0.060475038, -0.000851929341), G=(-0.0237213771, -0.000152844334) B=(B:-0.167404932, K:1.20047654), BG=(B:0.799203138, K:2.59157385), eta=0.268629073
25: K=(0, 0.0417413582), S=(0.0668472895, -0.000810970091), G=(-0.0229905809, -0.000227446401) B=(B:-0.173368521, K:1.18959885), BG=(B:0.733566497, K:2.76557016), eta=0.29549198
26: K=(0, 0.041228377), S=(0.0736408218, -0.000743712322), G=(-0.022194518, -0.000303446742) B=(B:-0.179404491, K:1.17683040), BG=(B:0.688563194, K:2.85241182), eta=0.325041178
27: K=(0, 0.0406961685), S=(0.0808549541, -0.000645079636), G=(-0.0213420481, -0.000380719616) B=(B:-0.185603421, K:1.1623433), BG=(B:0.648784148, K:2.8719165), eta=0.357545296
28: K=(0, 0.0401442549), S=(0.088485703, -0.000508955128), G=(-0.0204418951, -0.000459053316) B=(B:-0.192047025, K:1.14629892), BG=(B:0.618372935, K:2.84217945), eta=0.393299826
29: K=(0, 0.0395714544), S=(0.09655284968, -0.000328409539), G=(-0.0195023412, -0.000538127061) B=(B:-0.198802746, K:1.12883292), BG=(B:0.595228018, K:2.77826766), eta=0.432629808
30: K=(0, 0.0389868291), S=(0.104962711, -9.55997319E-5), G=(-0.0185311205, -0.00061749691) B=(B:-0.205920471, K:1.1100522), BG=(B:0.568193632, K:2.6917154), eta=0.475892789
31: K=(0, 0.0383931402), S=(0.113781817, -0.000198262595), G=(-0.0175354983, -0.000696590694) B=(B:-0.213431561, K:1.0900366), BG=(B:0.544543829, K:2.59046553), eta=0.523482068
32: K=(0, 0.0377866961), S=(0.122961136, 0.000562915332), G=(-0.0165224772, -0.000774732734) B=(B:-0.221349864, K:1.06884833), BG=(B:0.520402135, K:2.47923933), eta=0.575830275
33: K=(0, 0.0371797756), S=(0.132475279, 0.00100902899), G=(-0.0154990375, -0.000851152334), G=(B:-0.229673845, K:1.04665418), BG=(B:0.495535685, K:2.36066275), eta=0.633413302
34: K=(0, 0.0365762539), S=(0.142292576, 0.00154816111), G=(-0.0144723232, -0.000925008223), G=(B:-0.238388832, K:1.02317813), BG=(B:0.469134575, K:2.23605792), eta=0.696754633
35: K=(0, 0.0359822978), S=(0.152376234, 0.0021966529), G=(-0.0134449146, -0.000995421132), G=(B:-0.2474686062, K:0.99883463), BG=(B:0.441874516, K:2.1061524), eta=0.766430096
36: K=(0, 0.0354044143), S=(0.1626845, 0.002955586), G=(-0.0124387737, -0.00106149253), G=(B:-0.256875988, K:0.973612478), BG=(B:0.413633854, K:1.97155237), eta=0.843073105
37: K=(0, 0.0348492504), S=(0.1731171296, 0.0038505018), G=(-0.01174470889, -0.00113325187), G=(B:-0.266562744, K:0.947641185), BG=(B:0.384544672, K:1.83295969), eta=0.927380416
38: K=(0, 0.0343233451), S=(0.183787102, 0.0048913304), G=(-0.0104820544, -0.00117071122), G=(B:-0.27646909, K:0.921081014), BG=(B:0.354817878, K:1.69122839), eta=1.02011846
39: K=(0, 0.0338328418), S=(0.194480039, 0.00609208248), G=(-0.00955062865, -0.00122487961), G=(B:-0.286523431, K:0.894123931), BG=(B:0.324637915, K:1.54738293), eta=1.1221303
40: K=(0, 0.0333381577), S=(0.205197039, 0.00746655701), G=(-0.00865908655, -0.00126497225), G=(B:-0.296642487, K:0.866993244), BG=(B:0.294264736, K:1.40261474), eta=1.23443433
41: K=(0, 0.0329787158), S=(0.215885374, 0.00902796707), G=(-0.00781290837, -0.0012966147), G=(B:-0.306732035, K:0.839941534), BG=(B:0.263981122, K:1.25827183), eta=1.35777767
42: K=(0, 0.032622681), S=(0.226493407, 0.0107884816), G=(-0.007001599253, -0.0013191221), G=(B:-0.316688359, K:0.81324695), BG=(B:0.234098509, K:1.11583786), eta=1.49355549
43: K=(0, 0.0323164876), S=(0.2376972181, 0.0127586653), G=(-0.00627168788, -0.00133185518), G=(B:-0.326400548, K:0.787206861), BG=(B:0.204949115, K:0.97689397), eta=1.64291098
44: K=(0, 0.0320600245), S=(0.247276005, 0.014946783), G=(-0.0055815297, -0.0013342163), G=(B:-0.335753691, K:0.76212938), BG=(B:0.176878336, K:0.843100135), eta=1.80720207
45: K=(0, 0.0318514216), S=(0.257362957, 0.017359815), G=(-0.00494582063, -0.00132565064), G=(B:-0.344632994, K:0.738322313), BG=(B:0.150230408, K:0.716082069), eta=1.98792228
46: K=(0, 0.0316872118), S=(0.267194864, 0.0199932719), G=(-0.00436358476, -0.00130565853), G=(B:-0.352928727, K:0.716079883), BG=(B:0.125334558, K:0.597414687), eta=2.18671451
47: K=(0, 0.0315625938), S=(0.276786779, 0.0228483744), G=(-0.003833212157, -0.00127362598), G=(B:-0.360541807, K:0.695676768), BG=(B:0.102488047, K:0.488515418), eta=2.40538596
48: K=(0, 0.0314718126), S=(0.285955953, 0.0259121475), G=(-0.00335025873, -0.00122979057), G=(B:-0.367389677, K:0.677307339), BG=(B:0.0819385541, K:0.390564178), eta=2.64592456
49: K=(0, 0.0314086772), S=(0.294820424, 0.0291665847), G=(-0.00291252941, -0.00117376477), G=(B:-0.373411989, K:0.661160411), BG=(B:0.0638607835, K:0.304424846), eta=2.91051701
50: K=(0, 0.0313696881), S=(0.303297391, 0.032582847), G=(-0.00251573576, -0.00110575593), G=(B:-0.378575495, K:0.647316135), BG=(B:0.0483733657, K:0.230572267), eta=3.20156871
Loading time: 0:00:00.000
Processing time: 0:00:00.031
```

Функция ошибок в зависимости от $\vec{\delta}$

Успех поиска

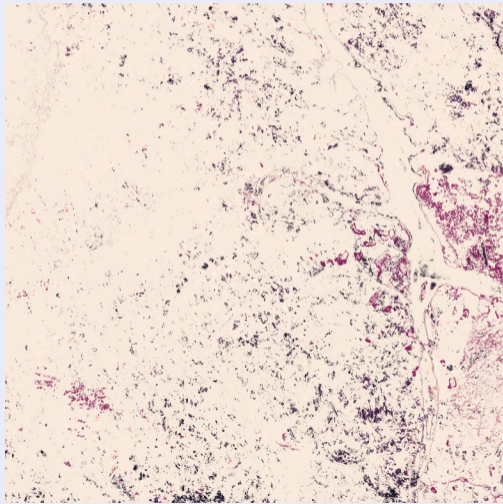


Ошибка поиска

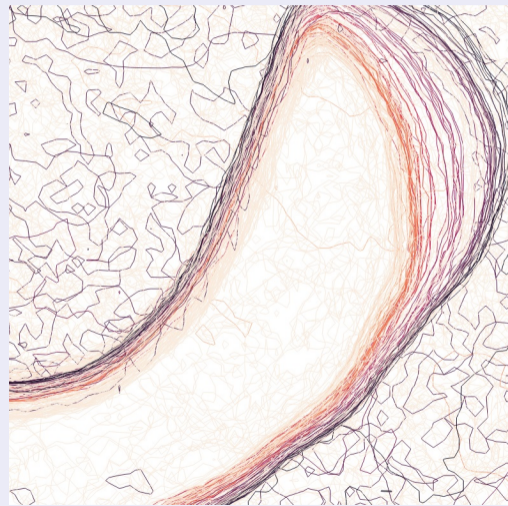


Все контура на уровне NDWI 0 до выравнивания, 2016-2024 гг.

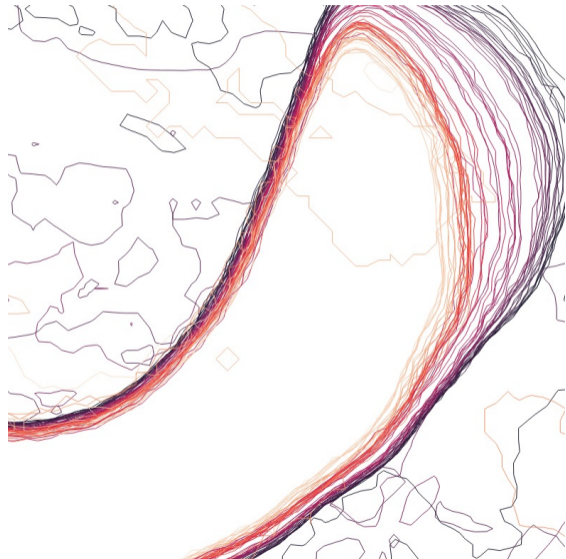
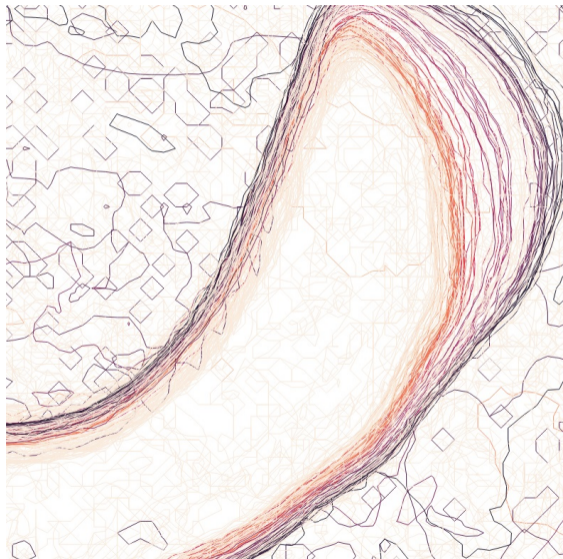
Вся трапеция



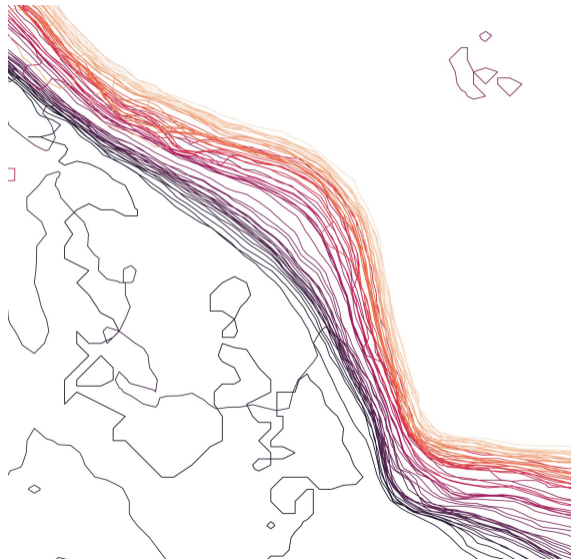
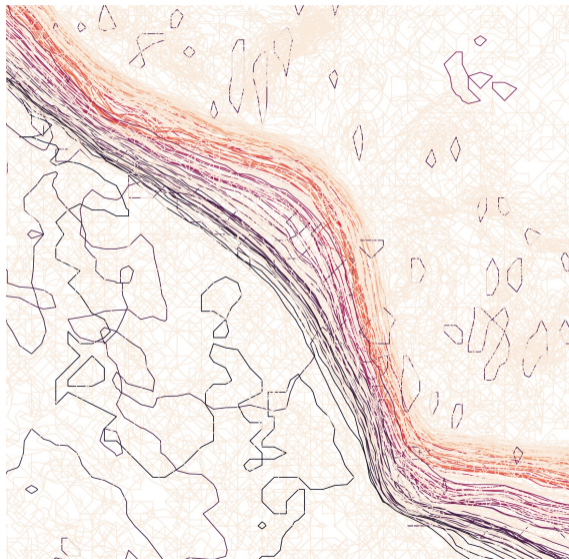
Фрагмент



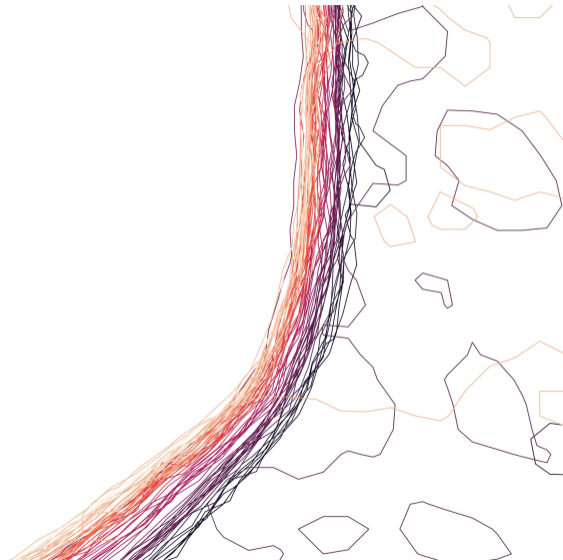
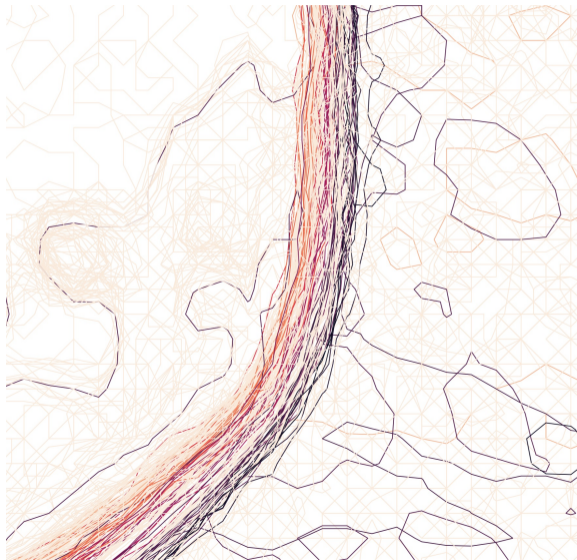
До и после выравнивания (1/4)



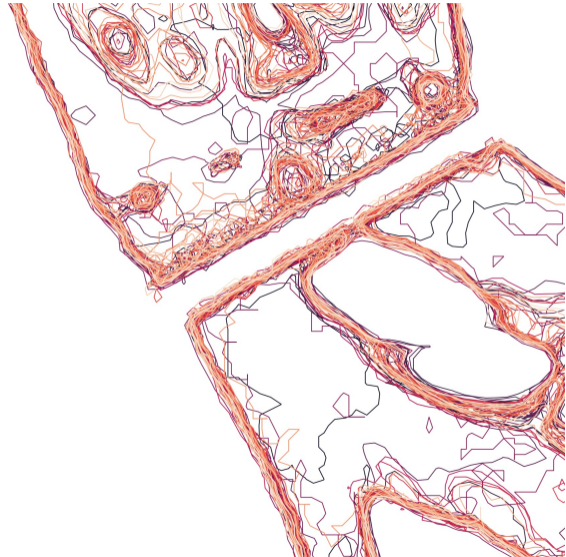
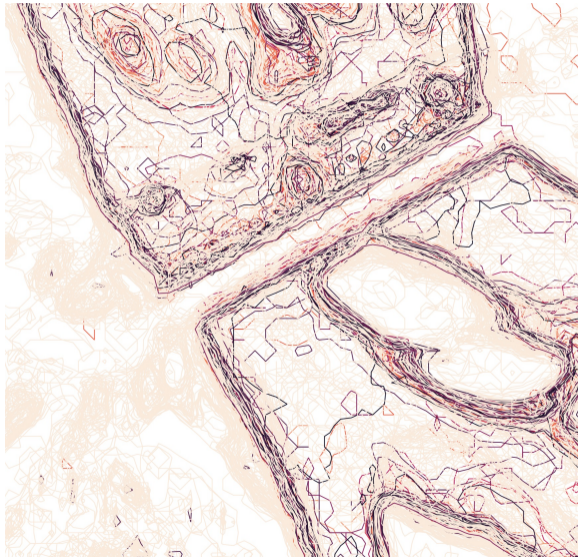
До и после выравнивания (2/4)



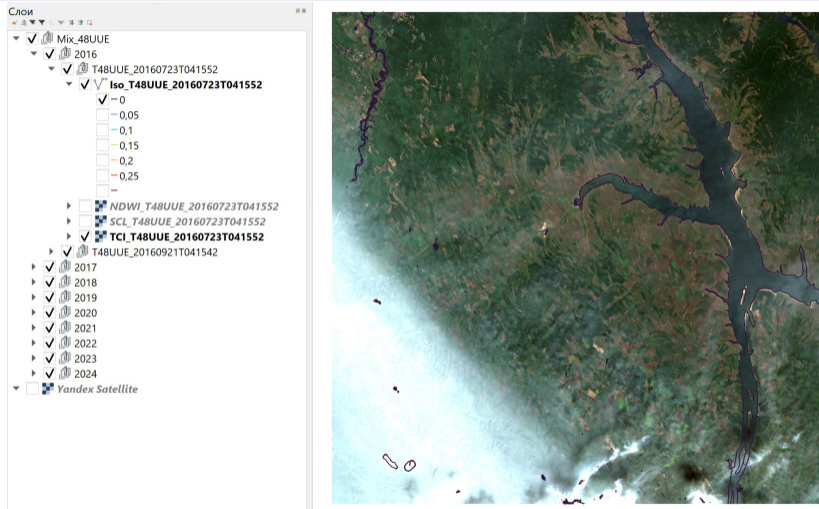
До и после выравнивания (3/4)



До и после выравнивания (4/4)



Редактируем вручную (без этого нельзя, но это быстро)

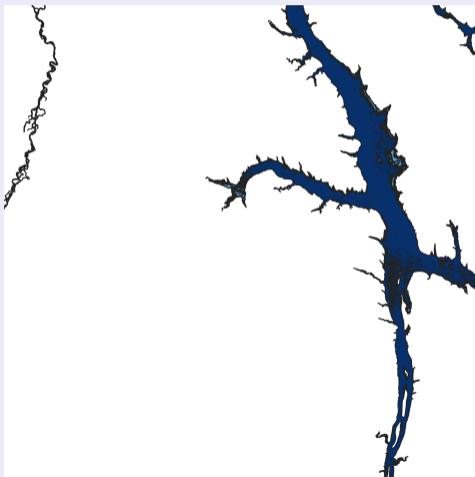


Сценарий на Python создаёт в QGIS дерево слоёв с загрузкой по необходимости. Удаляем вручную плохие слои (снег, лёд, облака) – группы слоёв целиком.

Результат редактирования

После удаления плохих слоёв (снег, лёд, облака), объединения оставшихся слоёв, удаления “мусора”

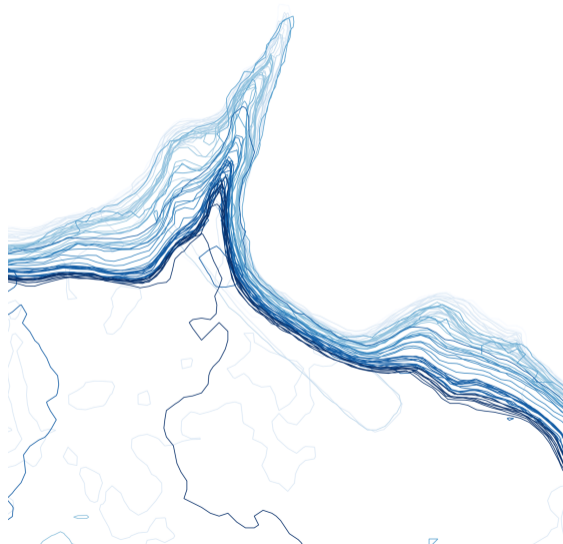
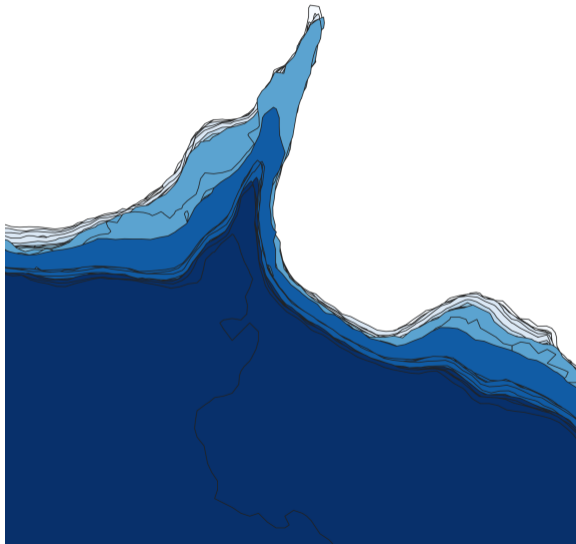
Вся трапеция



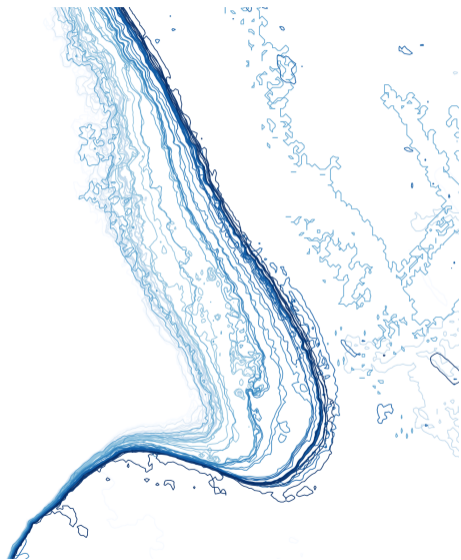
Фрагмент



После редактирования, более мелкий фрагмент



Ещё фрагменты после редактирования



Согласование неточных контуров

Контура для разных уровней могут пересекаться из-за погрешности измерений и влияния таких природных явлений, как волнение, облачность, и т.д.

Поэтому возникает задача построения модели рельефа по таким контурам береговых линий, получаемых с ошибками.

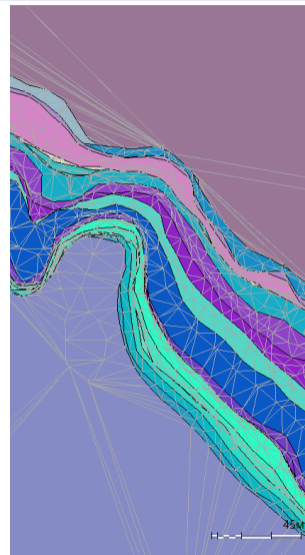
Требования к таким алгоритмам:

- 1 при отсутствии ошибок результат должен совпадать с триангуляцией, построенной по контурам, как по изолиниям рельефа
- 2 накопление информации (добавление новых контуров) должно повышать точность получаемой модели

Алгоритм согласования неточных контуров

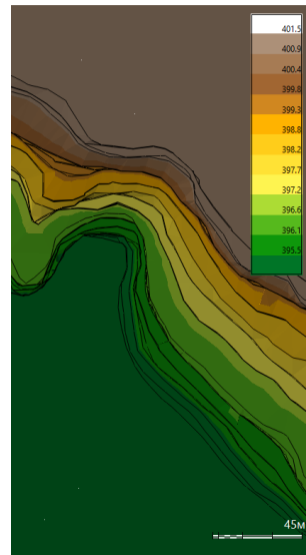
Основные принципы:

- Контура должны быть площадными, представляющими зоны затопления при соответствующих уровнях воды.
- Строим по ним вспомогательную триангуляцию Делоне с ограничениями, в которой линии контуров выступают в качестве жёстких рёбер.
- После включения контура в триангуляцию выполняем его заливку, которая состоит в том, что в каждый треугольник внутри контура добавляется информация о том, что он затапливается этим контуром.
- В результате для каждого треугольника триангуляции формируется список уровней, которые его затапливают.
- После завершения вспомогательной триангуляции каждому треугольнику сопоставляем истинный уровень его затопления – тот, который лучше всего согласуется со списком (если и для этого уровня получается много ошибок, то считаем, что треугольник не затапливается).
- Переносим в основную триангуляцию точки жёстких рёбер, разделяющих треугольники разных уровней, и сами эти рёбра. Высота ребра = \min уровень разделяемых им треугольников – соответствует первому требованию.

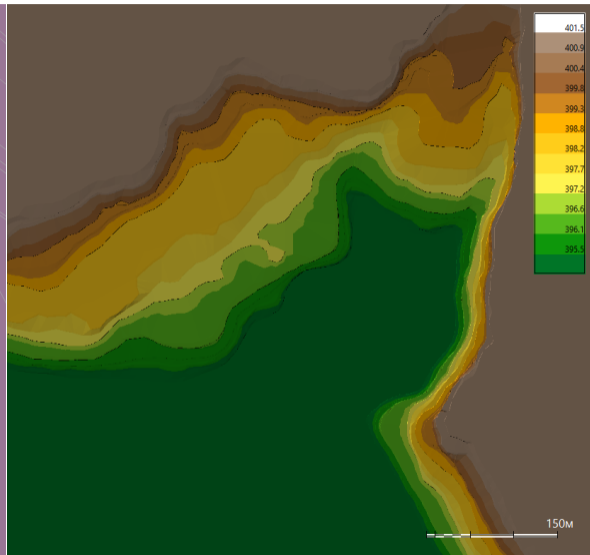
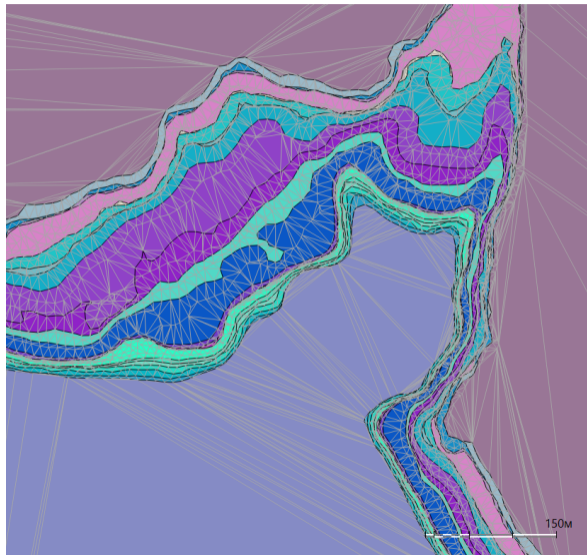


Нет времени подробно рассмотреть:

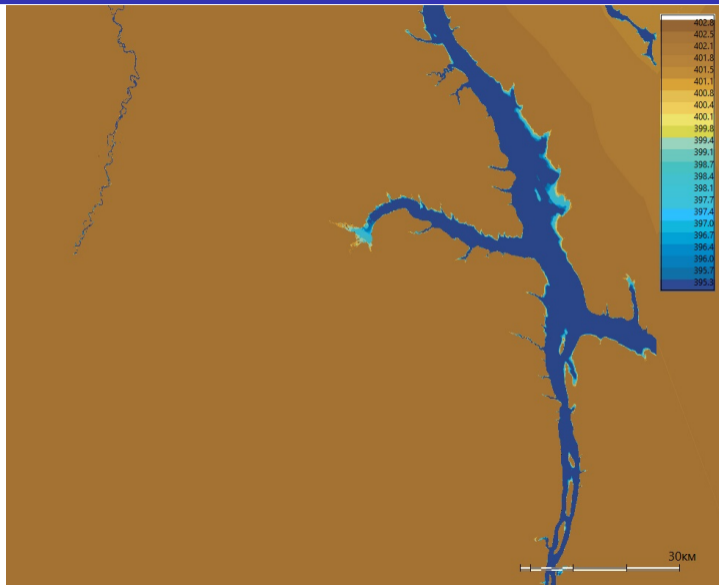
- Пока пропускались снимки с площадью покрытия менее 9900 км² – нужна модификация алгоритма согласования, учитывающая отсутствие данных.
- Вспомогательных триангуляций может быть несколько – для разных участков местности, т.к. данные некоторых снимков могут полностью отсутствовать на участке, что может ухудшить статистику, используемую при определении качества выбранного уровня.
- Триангуляция – это 2.5D, а не 3D: нельзя задать диапазон высот в точке. Поэтому ещё добавляем вспомогательные рёбра с небольшим смещением от настоящего положения ребра, если есть разрыв в уровнях разделяемых треугольников. При этом размер триангуляции возрастает примерно в 1.5 раза.



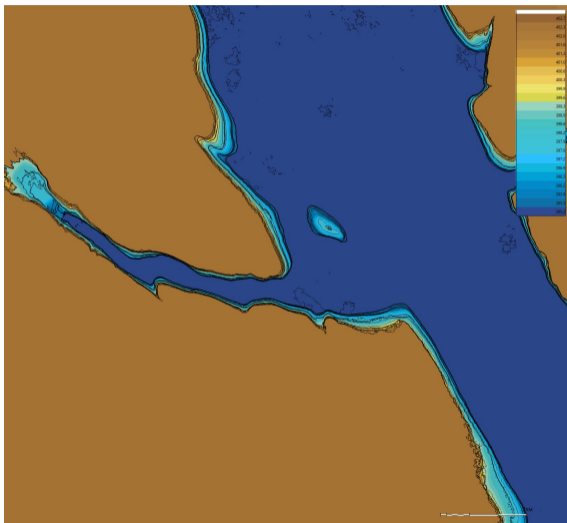
Сопоставление вспомогательной и результирующей триангуляций



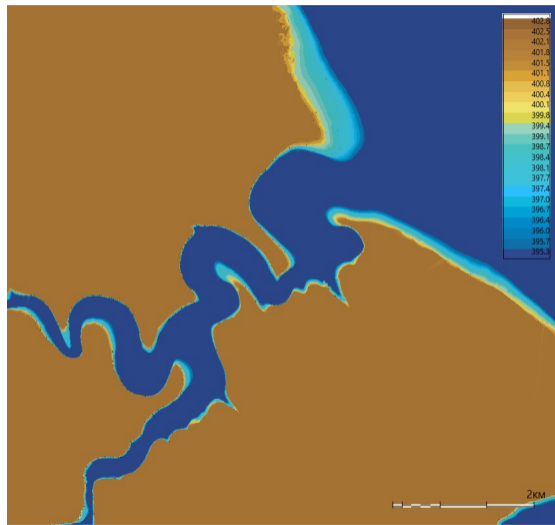
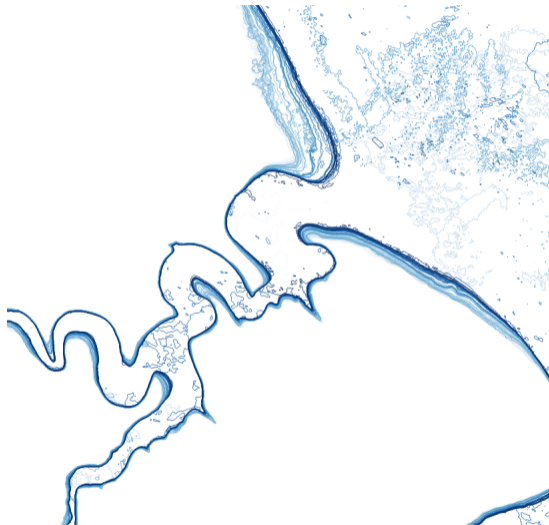
Вся результирующая триангуляция



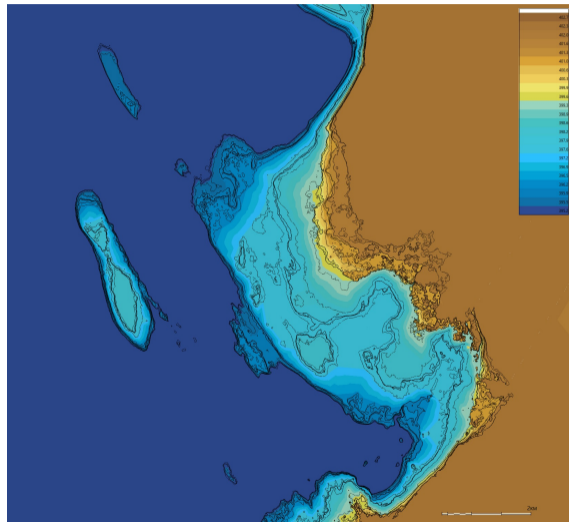
Сопоставление контуров и результирующей триангуляции (1/2)



Сопоставление контуров и результирующей триангуляции (2/2)



Сопоставление снимка Яндекс и результирующей триангуляции



Спасибо за внимание!

Нам было бы интересно применить описанную технологию к другим водным объектам с переменным уровнем воды.

Построение модели прибрежного рельефа в диапазоне наблюдаемых колебаний уровня воды по контурам береговых линий, выделяемых на спутниковых изображениях среднего разрешения

Алексей Евгеньевич Хмельнов

Институт динамики систем и теории управления имени В.М. Матросова
Сибирского отделения
Российской академии наук
(ИДСТУ СО РАН)
<http://idstu.irk.ru>

13 ноября 2024