Фиг. 1. Позволяваме обемен синтез на флуиди с висока разделителна способност и неразсейващи детайли с малък мащаб, използвайки CNN и хранилище за флуиден поток

Представяме нов алгоритъм, управляван от данни, за синтезиране на поток с висока разделителна способност симулации с многократно използваеми хранилища на данни за пространствено-времеви потоци. В нашата работа, ние използваме подход за учене на дескриптор, за да кодираме приликата между флуидни области с разлики в разделителната способност и числения вискозитет. Ние използваме конволюционни невронни мрежи за генериране на дескрипторите от флуидни данни като плътност на дима и скорост на потока. В същото време представяме деформация ограничаваща patch advection метода, който ни позволява стабилно проследяване на деформируеми флуидни области. С помощта на тази адвекция ние генерираме стабилни набори от пространствено-времеви данни от подробни флуиди за нашите хранилища.

Концепции за CCS: • **Изчислителни методологии** → **Физическа симулация**; *Процедурна анимация*;

Допълнителни ключови думи и фрази: симулация на флуиди, функция с ниски размери дескриптори, конволюционни невронни мрежи

1. ВЪВЕДЕНИЕ

Разрешаването на огромното количество детайли на естествените димни облаци е

дългогодишно предизвикателство за uid симулациите в компютърната графика.

Представянето на този детайл обикновено изисква много малки пространствени разделителни способности, което води до скъпоструващи симулации и от своя страна причинява болезнено дълго време на връщане. Разнообразие от мощни методи са разработени за облекчаване на този основен проблем: напр. подобряване на точността на стъпката на адвекция , последваща обработка на анимации с допълнителна синтетична турбуленция и ускоряване на стъпката на проекция на налягането.

Възприемаме различна гледна точка, за да реализираме ефективно потоците с висока разделителна способност: предлагаме да използваме голяма колекция от предварително изчислени пространствено-времеви региони, от които синтезираме нови с висока разделителна способност томове. За да намерите най-доброто съвпадение от това много ефективно хранилище, ние предлагаме да използваме нов, разбиращ потока дескриптор на функции. Ние ще гарантираме, че разстоянията L2 в това пространство на характеристиките ще съответстват до реални съвпадения на областите на потока по отношение на плътността на флуида, както и движението на потока, така че можем много ефективно да извличаме записи дори за огромни библиотеки от набори от данни за поток.

Тясно свързана с целта за ефективни потоци с висока разделителна способност е борба с числения вискозитет. Това е много труден проблем, т.к грешките в дискретизацията, възникващи в практическите настройки на потока, не могат да бъдат количествено изразени с изрази в затворена форма. На човешки везни вискозитетът на водата и въздуха е близо до нула, а ефектите са числени вискозитет бързо води до неестествено вискозни движения за практически резолюции. Докато ние не се стремим към метод, който директно намалява числен вискозитет, ще покажем, че е възможно да се предвиди влияние за типичните симулации на дим в графиката. Правим това в начин, управляван от данни, т.е., ние обучаваме модел да установява съответствие между симулациите с различни количества числови вискозитет.

В нашия метод, изчисляването на дескриптори и кодирането на ефектите от грешките на дискретизацията се обработват от конволюционен невронал мрежа (CNN). Тези мрежи се оказаха мощни инструменти за предсказване на сходството на двойките изображения , например, за изчисляване на оптичния поток или оценки на дълбочината. Ние използваме регресивните възможности на тези CNN обучават мрежи, които се научават да кодират много малък, но изразителен поток дескриптори. Тези дескриптори ще кодират съответствия в лице на грешки в числовите апроксимации и в същото време позволяват за много ефективно извличане на подходящи набори от пространствено-времеви данни от a хранилище.

Ние изчисляваме тези съответствия и търсения в хранилището локализирани региони в потока. Ще наречем тези региони кръпки в след това и ние използваме деформираща лагранжева рамка, за да проследим кръпки с течение на времето. В сравнение със записване на данни от статични или твърди региони, това има предимството, че малките характеристики са предварително изчислени и съхранявани в хранилището и не се разсейвайте по невнимание. В по този начин ние също така избягваме строгите ограничения във времето, които са глоби пространствената дискретизация обикновено налагат. От друга страна, ние трябва уверете се, че регионите не стават твърде лошо оформени с течение на времето. За това, ние предлагаме нова схема на адвекция, ограничаваща деформацията с стъпка на очакване. Мотивиран от фракталната природа на турбуленцията, и преобладаващо еднопосочен пренос на енергия към малки мащаби в енергийната каскада на Ричардсън , съвпадаме и проследявайте всеки пластир независимо. Това води до много ефективен метод, тъй като ни позволява да извършваме всички изчисления, базирани на корекция в успоредно.

***За да обобщим, приносът на нашата работа е:***

• нов, базиран на CNN подход за изчисляване на стабилни, нискоразмерни дескриптори на характеристики за плътност и скорост,

• деформация, ограничаваща адвекция на петна с очакване,

• алгоритъм за ефективен обемен синтез, базиран на региони пространство-време за многократна употреба.

В комбинация, нашият принос прави възможно много ефективно синтезират много подробни обеми на потока с помощта на повторно използваем хранилище за пространствено-времеви потоци. В същото време ние ще предоставим оценка на конволюционни невронни мрежи и обучение по сходство в контекста на базирани на плътност набори от данни за потока.

2 СВЪРЗАНА РАБОТА

Флуидните симулации за анимационни цели обикновено пропускат член на вискозитета на уравненията на Навие-Стокс (NS) и решаване на невисящи уравнения на Ойлер: Du/Dt = −∇p +g, където u,p и g обозначават скорост, налягане и ускорение, дължащи се съответно нa външни сили. Тези симулации имат дълга история в графиката , но особено за по-големи разделителни способности, изчисления поле на налягане, което прави потока без дивергенция, може да стане тясно място. Предложени са различни методи за ускоряване на необходимите изчисления, напр. груби проекции, мулти-мрежови решаващи задачи или с по-ниско измерение приближения. Друга важна част от Ойлериан изчислява транспорта в мрежата. Тук, безусловно широко се използват стабилни методи, особено полу-лагранжиан метод и по-точните му версии . Тъй като се фокусираме върху еднофазнитe потоци, ние ограничаваме след обсъждане на съответните произведения. За преглед на флуидни симулации в компютърната графика препоръчваме книгата от Р. Бридсън.

Докато алгоритмите за намаляване на алгоритмичната сложност са жизненоважни за бързите решаващи задачи изборът на структури от данни също е важен. Наред с другото, последните работи са предложили начини за обработка на повърхности, базирани на големи частици , или високоефективен дървесни структури.

Няколко произведения също са изследвали вискозитета за флуидни анимации, например чрез предлагане на стабилни и точни методи за дискретизация , ефикасни представяния на вискозните листи , или илюстриране на значението на вискозитета за разрешаване на срязващи потоци в близост до препятствия в потока . Въпреки че това са много интересни разширения, най-практични флуидни разтворители за компютърна анимация пропускат решаването на вискозитета в за да се намали изчислителната работа.

Методите, базирани на вихри, целят по-добро запазване на въртеливите движения на потоците, които обикновено са много важни за детайлите и реализъм. Предложени са методи за усилване на съществуващата завихряност , докато други произведения моделират ефектите на граничния слой или анизотропна вихри. Въпреки това, размерът на представителни подробностите все още са по своята същност ограничени от избраните разделителни способности на мрежата за тези алгоритми. Хибридни методи също са предложени за съчетайте Ойлерови симулации със симулации на лагранжева завихряност , докато други изследователи са предложили екстраполации на движенията на потока . Вместо това нашият подход разтоварва изчислителната тежест за производство на дребномащабни детайли до етап на предварително изчисление.

Така наречените техники с повишена резолюция, които увеличават видимата разделителна способност на полето на потока, са друг популярен клас алгоритми за симулиране на подробни потоци при умерени изчислителни разходи. Различен са предложени вариации за базирани на мрежа симулации , за мрежови ефекти на плаваемост или за симулация, базирана на частици . Докато тези произведения представляват изключително полезни алгоритми за генериране на подробни анимации, всички дребномащабни характеристики на поток все още трябва да бъдат разрешени и се отвежда по време на симулация. Това причинява значителни количества на изчислителна работа. Сложността на стъпката на адвекция с един от методите на полулагранжа е линейна по отношение на броя на неизвестните. По този начин, намаляването на размера на клетката от ∆x до ∆x/2 води до осем пъти повече пространствени степени на свобода. Освен това обикновено трябва да намалим съответно размера на времевата стъпка, за да предотвратим времето грешки при интеграция от доминиране на решението. Това означава там ние се сблъскваме с приблизително 16 пъти по-голямо натоварване, за да изчислим движението на адвектирани количества, като плътността на дима. За разлика от нашите методът напълно разделя фините пространствени мащаби с помощта на CNN-базирани дескриптори. По този начин нашият подход работи с предварително изчислени пространствено-времеви региони на реални потоци, вместо да се прибягва до процедурни модели. Като такива, ние избягваме рязкото нарастване на сложността в резултат на много фина пространствена дискретизация. Вместо това нашият метод работи само с флуидни данни с ниска разделителна способност и с ниско измерение дескриптори по време на време на симулация. Единствената операция с висока разделителна способност е синтезът на плътност, който може да се извърши на per основа на рамката по време на изобразяване, по тривиално паралелен начин.

Нашият метод споделя приликите с предишните подходи за синтезиране на текстури за анимирани потоци. Докато ранните произведения се фокусираха върху проблема с проследяването на данните за текстурата на течни повърхности , разнообразие от интересни алгоритми за синтезиране на двуизмерни течни текстури е разработена през годините , с различни допълнения и подобрения . Тези текстури подходите за синтез бяха разширени за работа със скорости на потока от Ma et al. . По-специално, това е единствената изпълнявана работа синтеза в три измерения, като цена за синтезиране на 3D обемите обикновено са значителни дори за умерени размери. За разлика от синтез на текстура, ние не генерираме директно висока разделителна способност изход, а по-скоро се съсредоточете върху ефективно и точно извличане подходящи набори от данни от голямо хранилище на предварително изчислени данни. Освен това нашият подход за машинно обучение ни дава свободата да кодират както сходството, така и физическите свойства в търсенето.

Подходът на щамповане, използван във филмовите продукции, по подобен начин използва повторно по-малки региони от предишни симулации . Този подход обаче обикновено неподвижно премества печатите, и не привежда съдържанието им в съответствие със симулацията. Ние открихме че контролируемата деформация е решаващ компонент, който трябва да се направи лагранжевото представяне на работата на кръпките. Подобна идея беше изследван по-рано за анимиране на двуизмерни потоци с честотно контролирани текстури . Докато алгоритъмът им използва мярка за размера на деформацията, за да се смеси с недеформирани съдържание, ние изрично изчисляваме ново деформирано състояние за нашите пачове който отчита както транспортирането на потока, така и нежеланото деформации. Това води до увеличен живот на регионите на пластира, и съответно намалява операциите на смесване.

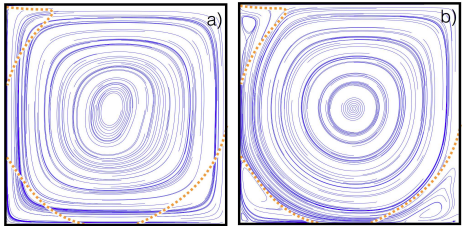
Машинното обучение с невронни мрежи е много успешно при различни предизвикателни проблеми с компютърното зрение. В по-специално, конволюционните невронни мрежи (CNN) са особено популярен , и няколко статии от тази активна област на изследване са насочени към изображението сходство, например за изчисляване на карти на дълбочината , или класификация на изображения с един кадър . Първият мрежи, използващи споделена клонова структура (т.нар. сиамски). предложено от Bromley et al. , докато се учи дескриптори с L2 разстояния бяха използвани за, например, дескриптори на човешки лица . В този контекст ще използваме вариант на популярната функция за загуба на шарнир, която представлява най-доброто изпъкнало приближение на бинарна загуба. Докато сходството на двойките изображения е изследвано преди, нашата цел е да работим с функциите на плътността и скоростта на флуида симулации. Ще демонстрираме, че е изгодно да плащате специално внимание към представянето на скоростта за учене. Досега съществуват много малко произведения, които комбинират алгоритми за машинно обучение с анимиращи течности. Изключение е подходът, базиран на регресия, за хидродинамика на изгладени частици. Този алгоритъм изчислява ускорения за лагранжиан NS Solver, базиран на внимателно проектирани входни функции, с цел да позволяват по-бързи симулации на течности. Нашият подход, от друга страна, има за цел автоматично извличане на възможно най-добрия набор от дискриминационни характеристики. Ние демонстрираме, че невронните мрежи могат да работят тази задача е много добре и за която можем от своя страна да използваме дескрипторите установяват съответствия между областите на груб и фин поток. Две други произведения споделят нашата цел да използваме невронни мрежи за еднофазни потоци: един фокусиран върху изучаването на проекцията на налягането стъпка с CNN и още едно обучение локални актуализации за премахване на дивергенцията. Особено първият от двата използва дълбока конволюционна мрежова структура това е подобно на нашето. Техният подход обаче се фокусира върху кодирането пълното поле на налягане на симулация на поток с невронна мрежа. В За разлика от тях, нашите мрежи научават стабилни дескриптори за по-малки региони на потока. Като такъв, нашият метод може лесно да се използва с произволно резолюции. И двата метода по-горе работят с пълната изходна разделителна способност и не са насочени към фините пространствени разделителни способности, към които се стремим нашия метод.

3 ПРИХОДСТВО НА ПОТОКА

Като се имат предвид две симулации на поток с един и същи ефект, като едната е грубо приближение, а другата е по-точна версия, напр. на по-фина пространствена дискретизация и пространствена област Ω, нашата цел е за да се изчисли резултат за сходство s за двата входа. Разглеждаме функциите Fc и Ff съответно на грубия и финия поток, където F може да бъде скаларна стойност, като например плътност на дима, или алтернативно може да включва и скоростта, т.е. F ∈ R 3 → R 4 . Ще разгледаме отново кои променливи на потока да включим в F в раздел 4.5, но засега ние може да приеме без загуба на общността, че F е скаларна функция. В за да изчислим сходството, трябва да извлечем достатъчно информация от област на потока, така че s може да заключи сходство от него. Ние ще пробва функциите на потока в редовна мрежа в рамките на Ω, ако приемем че F е достатъчно гладък, за да бъде представен от точкови проби. Всичко точкови проби от тази мрежа след това се комбинират във входен вектор xc и xf за груби и фини симулации, съответно

Като се имат предвид тези входни данни, ние се стремим да изчислим s(xc, xf ) за Ω такъв това s се доближава до нула, ако двойката всъщност е една, която съответства като едно и също явление е представено на грубо и фино везни. За увеличаване на несходството на потоците, s трябва да се увеличи. Това несходство може например да е резултат от разглеждането на различни региони Ω при фини и груби симулации или когато двете са изместени време.

Първото предположение би било използването на разстояние L2 за изчисляване на s като ∫ Ω kxf − xc k 2dx. Това се оказва неоптимален избор, тъй като дори малък преводите могат бързо да доведат до нежелано големи стойности на разстоянието.



Фиг. 2. Линии на потока на симулация на кухина, задвижвана от капак без (вляво) и с вискозитетът (вдясно) е показан. Оранжевите линии показват правилната позиция на централният вихър. В този случай графичният подход обикновено използван за изпускане на вискозитета води до много различна форма на вихър. По-лошо, при наличието на числен вискозитет, различни разделителни способности за Fc и Ff може бързо да доведе до значително различна скорост и съдържание на плътност, дори когато трябва да представляват едно и също флуиден поток. Фиг. 2 илюстрира колко силно вискозитетът може да повлияе на резултат от симулация. Обикновено се включва численият вискозитет на грешки във всички части на решателя, които влияят на скоростта (въпреки че стъпката на адвекция е може би най-големият принос). За практическите алгоритми не съществува решение в затворена форма, което да открие или измерете тези грешки. Вместо ръчно да се опитвате да намерите евристики или приближения за това как тези числови грешки могат да се разпространят и влияние върху решенията, ние прехвърляме тази задача към машинно обучение алгоритъм.

Освен това нашата цел е не само да измерим разстоянието между два входа, но по-скоро при нов груб вход искаме да намерим най-доброто съвпадение от голяма колекция от предварително изчислени набори от данни. Поради това ние картографираме проблема за съответствието в пространство на характеристиките, така че разстояния, изчислени с проста метрика на разстоянието, напр. Евклидов разстояние, съответства на желаното сходство s. Ще използваме нелинейна невронна мрежа, за да научим дескриптори на дискриминационни характеристики d(x) ∈ R m, като m е възможно най-малък. Като се има предвид област с груб поток Fc след това можем да извлечем най-доброто съвпадение от набор от фини региони Ff чрез минимизиране на ||d(xf ) − d(xc )||2

По-долу първо ще опишем нашия подход за дескриптори на потока на обучение с CNN. След това ще обясним нашите деформация наясно проследяване на кръпка, което представлява важна компонент, за да се постигне инвариантност по отношение на едромащабните движения. Накрая обясняваме как да синтезираме висока разделителна способност обеми за изобразяване.

4 ПРИХОДСТВО НА ПОТОКА НА УЧЕНЕТО

Тъй като нашият подход се фокусира върху изчисляване на разстоянията между потока входове с конволюционни невронни мрежи, ще разгледаме накратко най-важните подробности по-долу. На високо ниво можем да видим невронни мрежи като обща методология за апроксимиране на произволни функции f , където разглеждаме типични регресионни проблеми на форма y = f (x, w), т.е. при даден вход x, който искаме да приближим резултатът y е възможно най-близък при представяне, базирано на f на параметрите w. По този начин, за нашите проблеми с машинното обучение, компонентите на w са степените на свобода, които искаме да изчислим.

По-долу избираме невронни мрежи (NN), които да представят f , за които може да се намери задълбочен преглед, например в книгата на C. епископ. NN са представени от мрежи от възли, които, за разлика от нашето разбиране за природата, обикновено се моделират с непрекъснато функции. Ключова съставка са така наречените функции за активиране д, че въвеждат нелинейност и позволяват на NN да апроксимират произволно функции. За слой от възли L в мрежата, изходът на i’ти възел yi,L се изчислява с



Тук nL = означава броя на възлите в слой L и без загуба като цяло приемаме, че y0,L = 1. Тази константа вход за всеки възел моделира термин за отклонение, което е от решаващо значение за изместване на изхода на функция за активиране. Ние използваме тази често използвана формулировка да се слеят всички степени на свобода в w. По този начин, по-долу, ние няма да прави разлика между редовни тегла и пристрастия. Ние можем пренапишете уравнение (1) използване на матрица на теглото W като yL = g(WL−1yL−1). В това уравнение g се прилага по компонентно към входния вектор. Забележете, че без g бихме могли да сгънем цяла мрежа с нейното множество слоеве в една матрица W0, т.е., y = W0x, която може да бъде само използва се за изчисляване на линеаризирани приближения. Така на практика g е от решаващо значение за приближаване на общи, нелинейни функции. Типични избори за g са хиперболични допирателни или сигмоидни функции. За ученето процес, на мрежата обикновено се дава функция на загуба l(y, f (x, w)), който изчислява качеството на генерирания изход по отношение на y. Функцията на загуба трябва да бъде диференцируема, така че да е градиентна може да се разпространява обратно в мрежата, за да се актуализира тежести.

Ключов компонент, движещ много от последните успехи в дълбокото обучение, са така наречените конволюционни слоеве. Тези слоеве могат да се използват за използват пространствената структура на входните данни и по същество научават филтърни ядра за извивки. В областта на невронната мрежа тези филтрите често се наричат ​​карти на характеристики, тъй като те ефективно откриват последователни характеристики на входовете, които са важни за извеждане на желаното изход. Тази концепция е особено полезна за данни като звук (1D), или изображения (2D) и директно се простира до обемни набори от данни, като например като тези от флуидни симулации. Картите на характеристиките обикновено имат само малък набор от ненулеви тегла, например тегла 5×5 или 3×3. Като входните данни се състоят от векторни величини, например RGB стойности за изображения, невронните мрежи обикновено използват стекове от карти на характеристиките. следователно, можем да мислим за картите на характеристиките като тензори от по-висок порядък. За обемни входове с три пространствени измерения, съхраняващи векторна стойност функция, ние използваме тензори от четвърти ред, за да представим единична характеристика карта за конволюционен слой. Прилагането на тази карта на характеристиките дава а скаларна 3D функция. Повечето практични мрежи научават множество функции карти наведнъж на слой и по този начин генерира векторна стойност за следващ слой, чийто размер е равен на броя на картите на характеристиките предишния слой.

Тези конволюционни слоеве обикновено се комбинират с обединяване слоеве, които намаляват размера на слоя чрез прилагане на функция, като максимума, върху малка пространствена област. Ефективно това понижава семплирането на пространствено подредения изход на слой, по ред за да се намали размерността на проблема за следващия слой.

Имайте предвид, че конволюционните невронни мрежи по принцип не могат да направят повече от мрежи, състоящи се само от напълно свързани слоеве. Последните обаче обикновено имат значително по-голям брой степени на свобода. Това ги прави значително по-трудни за обучение, докато конволюционните мрежи водят до много разумни размери на мрежата и време за обучение, което ги прави много атрактивни на практика. По-малките мрежи също са значително намалени изисквания за количествата входни данни и може да бъде от полза за узаконяване. След това ще обясним предизвикателствата в нашата обстановка при избора подходяща функция на загуба за невронните мрежи и очертайте подробности за нашите мрежови архитектури след това.

4.1 Функции за загуба

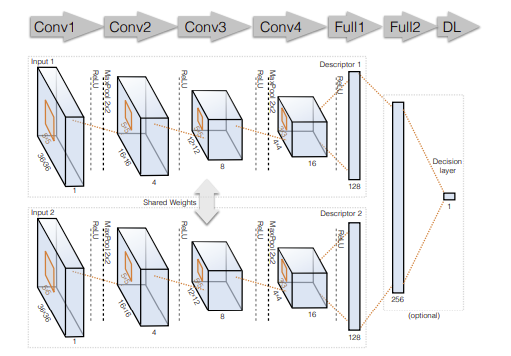
Когато изчисляваме нашата метрика за сходство на потока, първият подход за обучение може да бъде формулиран като s = fs (x1, x2, w), където fs отново е заучената функция, представена от w, x1 и x2, представлява двойка от входни характеристики, извлечени от две симулации, а s е изходът което показва колко сходна е входната двойка. Тъй като входните данни за нашите проблеми с регресията произтичат от хаотичен процес, т.е. турбулентен поток, входовете изглеждат „шумни“ от гледна точка на регресията и обучението наборът от данни обикновено не е линейно разделим. От решаващо значение е ученето процес не само кодира понятие като Евклидово разстояние на двата входа, но също така се научава да използва цялото представено пространство чрез неговата мрежова структура, за да се изчисли сходството. В същото време, мрежата трябва надеждно да открие различни двойки. следователно, наличието на отрицателни двойки е от решаващо значение за установяване на стабилност сходство между истински позитиви.

Основният проблем на подобието на обучението може да бъде формулиран по следния начин: като се даде двойка входни данни, генерирайте етикет y ∈ {1, −1}, т.е. двойката е подобен (1) или не (−1). Докато е наивна L2 загуба да научите точно тези етикетите е очевидно недостатъчна, може леко подобрена функция за загуба се формулира като ln = −y f (x1, x2, w). В този случай мрежата ще бъдат възнаградени, за да разделят положителни и отрицателни двойки, но поради липсата на каквато и да е граница, научените стойности биха се отклонили до плюс и минус безкрайност. Вместо това е от решаващо значение да имате функция на загуба, която не ограничава ненужно регресора, и в същото време му дава свободата да натиска правилно класифицирани двойки, колкото е необходимо. Установената функция на загуба в тази настройка е така наречената загуба на шарнир, която може да се изчисли с:



Тази функция на загуба обикновено води до значителни подобрения над наивните функции на загуба, описани по-горе. Когато се използва NN представяне във връзка с функцията на загуба на уравнение. (2), характеристика дескрипторът може да бъде извлечен чрез използване на изходите на последния напълно свързан слой като дескриптор

Докато този подход работи, ние ще демонстрираме, че е още по-добре да вградите разстоянието L2 на дескрипторите директно в пантата загуба . Тъй като по-късно базираме нашите търсения в хранилището на тези разстояния е важно да се насочи NN към кодиране дискриминационни разстояния въз основа на самите дескриптори на характеристиките, вместо да се оптимизира само за краен резултат за сходство s. За да направим това, можем да преформулираме учебния проблем, за да генерираме самият дескриптор, използвайки разстоянието на дескриптора като "несходство".



Фиг. 3. Нашата CNN архитектура с две конволюционни стека със споделени тегла, последвани от функция и незадължителен слой за решение.

По-долу ще обозначим изходите на конкретен дескриптор слой на нашата мрежа с dw (x), където x е входът, за който да изчислете дескриптора. Въз основа на тези дескриптори променяме проблема за регресията на fe (x1, x2, w) = β −α ||dw (x1) −dw (x2)||, α > 0. Тук сме въвели параметрите α и β за фина настройка на начало и стръмност на функцията. Използването на fe вместо fs в уравнение. (2) добиви



където сме заменили параметрите α, β с αp,n, което може да бъде използва се за фина настройка на полетата поотделно за положителни и отрицателни двойки, както ще обсъдим по-долу. Имайте предвид, че нормализираме дескрипторите dw, когато ги извличаме за уравнение. (3). Това значително се подобрява конвергенция и поддържа разпределението на компоненти за обучение, вместо абсолютни стойности в дескриптора. Ще демонстрираме че тази функция на загуба явно превъзхожда другите алтернативи, след като описваме подробностите за нашите невронни мрежи, в които да се използва във връзка с тази функция на загуба.

4.2 Архитектура на CNN

Нашите невронни мрежи се състоят от типичен стек от конволюционни слоеве които превеждат пространствено подредените данни във все по-голям брой на сигнали с по-ниска пространствена разделителна способност. Визуално обобщение на мрежата е дадена на фиг. 3. Тъй като нашата мрежа сравнява две входове, той първоначално има два клона, които съдържат дублиран стек от конволюционни слоеве със споделени тегла. Всеки клон действа отделно на един входен вектор, за да се намали неговата размерност. В изходи на последния конволюционен слой на всеки клон (Conv3 in нашия случай) са свързани в сериен вектор, съдържащ двете дескриптори на функции (слой Full1). За мрежи, които регресират единична оценка на сходство с функция на загуба като уравнение. (2), добавяме още един напълно свързан слой (Full2) и изходен слой с един изход възел, който изчислява крайния резултат за сходство. Тези двамата обаче слоевете са по избор. Когато тренирате със загуба на вграждане на пантите, уравнение (3), пропускаме тези два слоя.

Ще използваме следните съкращения, за да посочим мрежата структура: конволюционен слой (CL), максимален обединяващ слой (MP) и напълно свързан слой (FC). Започваме с входове с размер 36x36 в 2D. Входът от симулация с ниска разделителна способност се увеличава линейно до тази резолюция (обикновено използваме четири пъти по-ниска разделителна способност за грубата симулация). Един конволюционен клон на нашата мрежа добиви 2 2 точки с по 32 характеристики всяка, т.е. общо 128 стойности. Тези 128 изхода са обединени в крайния слой с дескриптор на характеристики dw с нормализиране. За триизмерни входове ние разширяваме пространственото измерение във всеки слой съответно. Следователно, на първият слой има разделителна способност 5x5x5x4 в 3D, а крайният пространствен резолюцията е 2 3 с 32 функции. По този начин в 3D, нашият дескриптор на функции Dw има измерение 256.

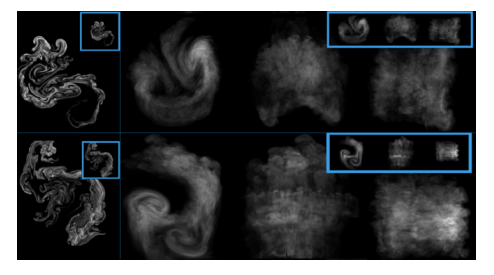
4.3 Генериране на данни

За подходите за машинно обучение е от решаващо значение да имате добро обучение набори от данни. По-конкретно, в нашата среда предизвикателството е да създадем контролирана среда за симулации с различни дискретизации, и което води до различни числени вискозитети. Без специални грижи, грубите и фините версии ще се отклонят с течение на времето и поради тяхното нелинейност, първоначалните малки разлики бързо ще доведат до напълно различни потоци. По-долу разглеждаме сходството на потока вътре избран времеви хоризонт tr . Имайте предвид, че този параметър зависи от разглеждана настройка, например за много плавни и бавни движения, по-големи стойности са подходящи, докато бурните и бързи движения означават потоци се разминават по-бързо. Ще обсъдим последиците от избора на tr in повече подробности по-долу.

Ние използваме рандомизирано първоначално условие, за да създадем нашите данни за обучение. Като се има предвид първоначално условие, ние създадохме две паралелни симулации, едната с груба разделителна способност на rc клетки на ос и обикновено използваме a четири пъти по-висока разделителна способност rf = 4rc за фината версия. Докато е би било възможно просто да се стартира голям брой симулации за a time tr , открихме, че е за предпочитане вместо това да се синхронизира симулация в интервали с дължина tr . Тук ние даваме приоритет на високото разделителна способност, като се приеме, че неговият по-нисък числен вискозитет е по-близък към истинското решение. По този начин отново инициализираме грубата симулация в интервали с дължина tr с нискочестотна филтрирана версия на фината симулация.

Тази синхронизация води до различни интересни и разнообразни конфигурации на потока във времето, които иначе бихме имали да пресъздаде ръчно с различни начални условия. За нашите данни поколение, установихме, че плаващите потоци са проблематични в правоъгълни домейни поради тяхното възходящо движение. Използването на високи домейни би от курсова работа, но обикновено губи значително количество място. Вместо това, ние изчисляваме център на масата за плътността на дима по време всяка времева стъпка. След това добавяме корекционен вектор по време на адвекцията на полуЛагранж за всички количества, за да преместим центъра на масата до центъра на мрежата. По време на генерирането на данни ние зареждаме пачове в целия обем и ги проследявайте със същия алгоритъм, който ние използвайте за синтез по-късно. По този начин ние също използваме нашата деформация ограничаване там, което описваме подробно в раздел 5.1. За всеки област на кръпка, ние записваме пълната груба и фина скорост/плътност функции в рамките на всяка деформираща област на кръпка за всяка времева стъпка. В момента проследяваме кръпките с фината симулация и използваме същата пространствена област в грубата симулация.

Записаните двойки пространствени области за една времева стъпка ни дават набор от положителни двойки за обучение. Имайте предвид, че груби и фини данни в тези региони може да са се отклонили до продължителността tr . Да създам отрицателни двойки, ние присвояваме произволен набор от фини данни на всеки груб вход. Две характеристики от отрицателна двойка се записват или от различни пачове, или записани от същия пластир, но в различни времеви стъпки. Следователно, за всеки пример за груба характеристика x1 в нашето обучение и набор от данни за оценка, има само един пример за фина характеристика x2 маркиран като релевантно.



Фиг. 4. Примери за нашето генериране на данни за обучение, както в 2D (вляво), така и в 3D (изглед отпред, отляво и отгоре). Грубата симулация (със син контур) е синхронизиран с данните с висока разделителна способност на интервали tr .

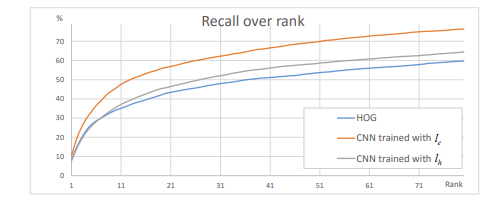
По този начин създадохме няколко комбинирани симулации и в двете 2D и 3D, с tr = 20 и tr = 40, за генериране на набори от данни за обучение както и набори от данни за оценка. Тези tr са избрани така, че получените данни за обучение имат максимални възможности за дискриминация. За по-малки интервали мрежата вероятно вижда само много подобни входове и следователно не могат да генерират експресивни дескриптори. Кога интервалът става твърде голям, входовете могат да станат твърде различни за NN. Като цяло, tr е отрицателно свързано с времевата стъпка, кинетична енергия и разлика в разделителната способност. В момента избираме tr ръчно чрез сравнения и автоматично изчисляване на tr остава интересна бъдеща посока.

Могат да бъдат намерени няколко изображения на нашето генериране на данни в 2D и 3D на фиг. 4. Подробните симулации имат 4 пъти по-висока разделителна способност. За обучение генерирахме 18 449 положителни двойки за 2D и 16 033 двойки общо за 3D. В 2D всяка тренировъчна партида съдържа 1:1 позитив и отрицателни двойки. Последните се генерират на случаен принцип от всички положителни по време на тренировка. Докато съотношението 1:1 беше достатъчно 2D, обучението с това съотношение се оказа бавно в 3D. Ние открихме че увеличаването на броя на отрицателните двойки подобрява ученето ефективност, като има незначително влияние върху конвергентираното състояние на CNN. По този начин използваме съотношение 1:7 за 3D тренировъчни бягания. За нашите оценки по-долу, наборите от данни с tr = 20 и tr = 40 дават много сходни резултати. Тъй като последният показва по-ясни разлики между методите, в раздел 4.4, ще фокусираме нашите оценки върху набора от данни с tr = 40, който има 5440 и 5449 положителни двойки в 2D и 3D съответно.

4.4 Оценка

За да се оцени и сравни успехът на различните подходи, би било направо да се изчислят дескриптори с избран метод за грубо въвеждане i и след това намерете най-доброто съвпадение от голяма колекция от фини двойки. Ако най-доброто съвпадение е това първоначално съответстващ на i, можем да смятаме това за успех, и провал в противен случай. По този начин можем лесно да изчислим процент от успешно извлечените двойки. Този показател обаче не би представят добре настройката на нашето приложение. Нашата цел е да използваме дескрипторите за кръпки в нови симулации, които нямат идеално съответстващ в хранилището. За тях искаме силно извлечете най-близкото налично съвпадение. Така, вместо да броим перфектни съвпадения, искаме да оценим колко надеждни са нашите мрежи са се научили да кодират сходството на потока в дескриптора пространство. За да определим количествено тази надеждност, ще използваме следната мярка истинският положителен процент, който обикновено се нарича изтегляне, над границата ранг k.

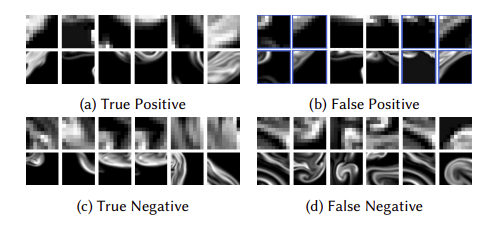
Извикването върху ранг на прекъсване обикновено се използва в полето за извличане на информация за оценка на резултатите от класираните резултати. Recall означава процентът на правилно извлечените набори от данни за всички дадени свързани. Рангът в този случай показва броят на най-близките съседи, които са извлечени от хранилището за даден вход. По-специално, за нашия набор от данни за оценка с N двойки, с дадена граница k, ние оценяваме извикването за всички N груби характеристики и по този начин kN двойките се извличат общо за оценка. В тези извлечени двойки, ако r двойки са правилно обозначени като свързани, the припомнянето при прекъсване k би било r/N. В такъв случай перфектен метод, ще даде 100% за изтеглянето при k = 1 и след това ще бъде постоянна за по-голям k. На практика методите бавно ще се приближават до 100% за увеличаване на k и дори най-лошите методи ще постигнат извикване от 100% за k = N. По този начин, особено първият диапазон от малки k стойности е интересен за да се оцени колко надеждно даден метод е успял да групира подобни данни в евклидовото пространство на стойностите на дескриптора. Първо сравняваме два базирани на CNN дескриптора, създадени с две функции за загуба, обяснени по-горе, и популярната, ръчно изработена HOG дескриптор в 2D. Последният е често използван и многоуспешен дескриптор на характеристики, базиран на хистограми на градиенти на a местен регион. Както може да се види на фиг. 5, дескрипторът HOG се представя най-зле в тази настройка. Отвъд ранг от 6, припомнянето е очевидно под това на обикновения CNN дескриптор за загуба на шарнир. В пантата е вградена загуба функцията дава най-добрия дескриптор, което в този случай е последователно повече от 10% по-добре от ранг 10 нататък. Високите нива на припомняне показват че нашият CNN успешно се научава да извлича дискриминационни характеристики, и може да направи това с по-висока точност от конвенционалните дескриптори.



Фиг. 5. Извикване над ранга за HOG (синьо), CNN обучен с lh по сходство изходен възел (сив) и CNN, обучен с le на дескриптори директно (оранжев).

Ние също така изследваме влиянието на прага αp и αn във функцията на загуба le в уравнение. (3). Тъй като параметризацията [0.0, 0.7] има малко по-висока точност сред другите, ще използваме тези параметри по-долу. Установихме, че не е необходимо имат марж от положителната страна на функцията на загуба le , но от отрицателна страна, относително голям марж дава на нашия CNN способност за по-добро научаване на различията на двойките.

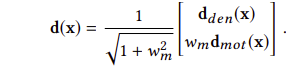
Фиг. 6 показва някои от най-добрите верни и неверни съответствия, направени от нашия CNN за двойките плътност на дима. Вляво са показани правилните положителни и отрицателни двойки. Фалшиво отрицателни двойки са свързани, за които дескрипторите на CNN все още имат голям разлика, докато фалшиво положителните се съпоставят погрешно двойки, които не са свързани. На практика фалшиво отрицателната двойка нямат ефект върху синтезираните резултати, за разлика от фалшивите позитиви. Въпреки това забелязваме, че тези фалшиви положителни резултати обикновено съдържат визуално много сходни данни. Като такива, тези набори от данни ще бъдат малко вероятно е да въведе визуални артефакти в крайния том. Някои от тези фалшиви положителни резултати всъщност произлизат от един и същ проследен регион на корекция по време на генерирането на данни и са класифицирани само погрешно в условия на съвпадащото им времево разстояние. Тези двойки са маркирани с сини граници на фиг. 6(b).



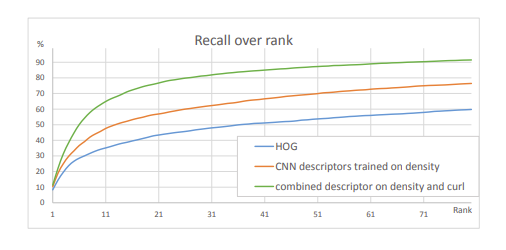
Фиг. 6. Най-високо класирани двойки плътност, съпоставени от нашия CNN

4.5 Дескриптори за движения на потока

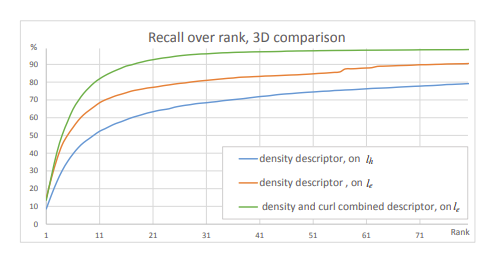
Досега ние разглеждахме само плътността на дима за установяване съответствия, обаче, в контекста на симулация на флуиди ние също има информация за скоростта. Скоростите силно определят движението на дима с течение на времето и като такива те също са важни за създаване на съответствия между данните от нова симулация и наборите от данни в хранилището. За да стигнем до метод, който също взема предвид движението на потока, ние използваме две мрежи: едната е обучена специално за дескриптори на плътност, докато обучаваме втората специално за движението. Това дава два дескриптора, dden и dmot, в които свързваме един дескрипторен вектор за търсене в нашето хранилище с



Имайте предвид, че отделното изчисляване на плътността и дескриптора на движение означава, че можем лесно да променим мащаба на двете половини, за да поставим повече акцент на едното или на другото. Особено при синтезиране на нови симулации резултати, ние поставяме повече акцент върху съдържанието на плътност с wm = 0,6. За дескриптора на движение използваме завихрянето като вход, т.е. ω = ∇ ×u. По време на стъпката на синтез, дескриптори на движение, генерирани от vorticity предлагат значително по-добри прегледи от обучените



Фиг. 7. Използването на дескриптори на къдрици, както и на дескриптори на плътност подобрява производителността на съвпадение.



Фиг. 8. В 3D използването на дескриптори на къдрици, както и на дескриптори за плътност, подобрява съвпадаща производителност.

с u, тъй като завихрянето отразява по-добре локалните промени в полето на потока. Поради нашето разделяне на мащаба с движение на кръпка и съдържание, нашата целта е да представим локални, относителни движения с нашите дескриптори, вместо, например, широкомащабни преводи или ефекти на разтягане. Използването на a комбинираната плътност и дескриптор на къдрици подобрява дори скоростта на припомняне по-нататък. Сравнение с нашия набор от 2D данни е показано на фиг. 7, напр. на ранг 11, припомнянето се подобрява с около. 35% и виждаме подобни печалби в три измерения (показани на фиг. 8). Имайте предвид, че тези две фигури използвайте тегло от wm = 1.0 за дескриптора на къдрици.

Поради гореспоменатите подобрения в точността на съвпадение, този подход представлява нашия краен метод. По-долу ще използвайте двойна мрежа, една обучена за плътности и втора обучени за curl да изчислява нашите дескриптори

4.6 Директен синтез на плътност

Виждайки генеративните възможности на съвременните невронни мрежи, ние намери за интересно да проучи колко далеч могат да бъдат тези мрежи изтласкани в контекста на потоки с висока разделителна способност. Вместо да се стремим към изчисляването на дескриптор с ниска размерност, NN също може да бъде получи задачата да регресира директно петно ​​дим с висока разделителна способност плътности. Установена мрежова структура за тази цел е стек на конволюционни слоеве за намаляване на входния регион до по-малък набор на функции за реакция на функции, които след това задвижват генерирането на изходът със стек от слоеве за транспониране на навиване



Фиг. 9. Директният опит за синтезиране на плътности с CNN дава замъглен резултат резултати, при които липсват структури (вляво). Нашият алгоритъм изчислява много детайлно потоци за същия вход (вдясно).

Проведохме обширна серия от тестове и най-добрите резултати, които можехме постигане за директен синтез на плътност са показани на фиг. 9. В това в случай, мрежата получава област от 16x16 стойности на плътност и произвежда резултати с четири пъти по-висока разделителна способност (64x64) с помощ от два конволюционни слоя, напълно свързан слой и четири деконволюционни слоеве. Докато бихме могли да осигурим сближаване на мрежи без преоборудване и относително добра времева стабилност, на синтезираните плътности липсват детайлни структури. Тази липса на детайлност възниква въпреки факта, че тази мрежа има значително по-голям брой тегла от нашата дескрипторна мрежа и имаше повече данни за обучение на негово разположение.

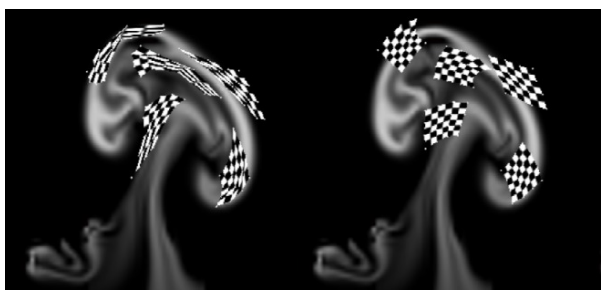
Очевидно няма доказателство, че е невъзможно да се синтезират детайли плътности на дима с генеративни невронни мрежи, но ние вярваме че нашите тестове илюстрират, че хаотичните детайли на бурния дим потоците представляват изключително предизвикателна задача за NN. Особено когато се опитвате да избегнете прекомерното оборудване с достатъчно голям брой на входове, турбулентните движения изглеждат като шум за мрежите. В резултат на това те научават усреднено поведение, което изглажда подробни структури. Тези резултати също мотивират нашия подход: ние избягваме тези проблеми, като се научим да кодираме разстоянието между регионите на потока и предоставяне на най-добре съвпадащи детайли в нашите flow хранилище по време на изобразяване, вместо да учи и генерира подробности директно

5 ДВИЖЕНИЕ И СИНТЕЗ

За всеки пластир проследяваме движението му с локална мрежа. Ние наричаме тези решетки клетки по-долу, за да ги разграничат от картезиански решетки на симулациите на флуида и ще обозначим броя на подразделенията на пространствена ос с ncage, с резултантния размер на клетката ∆xcage. По-долу описваме нашия подход за ограничаване на прекомерната деформация на тези клетки.

5.1 Движение, ограничаващо деформацията

Дори простите потоци показват големи количества въртеливо движение, това бързо водят до много нежелани артефакти на разтягане и инверсия. Пример може да се види от лявата страна на фиг. 10. По този начин, когато отклоняваме областите на Лагранжево петно ​​през поле на скоростта, ние сме изправени пред предизвикателството да избягват прекалено силни изкривявания по време на изработката пластирът следва предписаното движение. Вдъхновен от предишна работа при възможно най-твърди деформации, ние изразяваме нашите лагранжеви клетки чрез диференциал координати и минимизиране на енергийния функционал в най-малките квадрати смисъл да се извлече конфигурация, която ограничава деформацията, докато се придържа към движението на потока.

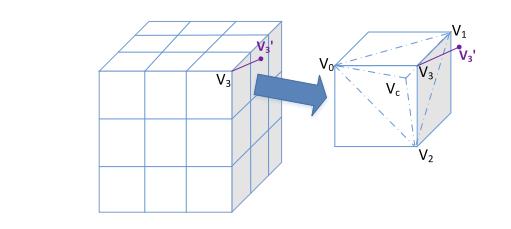


Фиг. 10. Наивната адвекция на пластира (вляво) бързо води до изкривени области. Нашата адвекция, ограничаваща деформацията (вдясно), може да запази стабилни региони докато след движението на течността.

За диференциалните координати обхващаме въображаем тетраедър между произволен връх v3 и неговите три съседни върха v0,1,2, както е показано на фиг. 11. За недеформираното състояние на клетка, позицията на v3 може да се изрази с завъртания на тетраедъра ръбове като

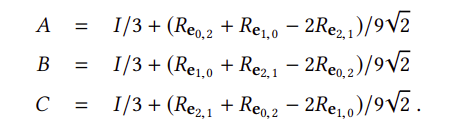


Тук ei,j означава ръба между точките i и j, а Rv е точката 3x3 ротационна матрица, която се върти на 90 градуса около оста v. vc е геометричният център на триъгълника обхващаше трите съседи, т.е. vc = (v0 + v1 + v2)/3.



Фиг. 11. Пример за кръпка клетка с n = 3. Грешката при деформация е натрупани от всяка клетка, както е показано вдясно. Разстоянието между целева позиция v3 и адвектирана позиция v 0 3 води до деформация грешка. v3 се изразява чрез v0,1,2 и тяхната централна точка vc.

Можем да пренапишем уравнение. (5) като v3 = Av0 + Bv1 + Cv2, където

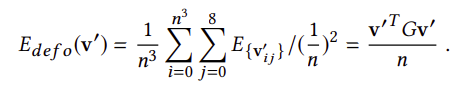


За нова длъжност на v 0 3 , напр. по-късно по време на симулация, можем

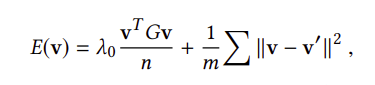
измерете квадратната грешка с



Съответно можем да изчислим обща грешка на деформацията за цялата клетка с m = (n + 1) 3 нови позиции v 0 с



За цяла лепенка клетка с n 3 клетки, натрупваме деформацията енергия за осемте ъгъла във всяка клетка. Енергията за сингъл връх е даден от E{v 0 i j } по-горе, където i и j са индексът на клетката и нейния ъгъл съответно. Дясната страна на уравнение (7) е а съкратена нотация, където G е 3m × 3m матрица, съдържаща натрупани вноски за всички точки на клетката. Тъй като всеки връх има най-много 6 свързани съседи, всеки ред вектор в G има at най-много 19 записа, съответстващи на позициите x, y и z на неговите съседи и диагонален вход за себе си. Минимизиране на тази квадратична форма директно ще доведе до тривално решение на нула, така че е необходимо за решаване на този проблем с подходящи ограничения. На практика искаме решението за зачитане на отклонените позиции. За това добавяме an допълнителна грешка на адвекция kv – v 0 к 2 което дърпа върховете към адвектираните позиции, т.е. v 0 . По този начин общата енергия, която свеждаме до минимум е:



където v 0 е адвектираната позиция, m = (n + 1) 3 е броят на върхове в мрежата, а λ0 контролира баланса между адвекцията и деформация. Имайте предвид, че в оригиналната формулировка Rv и така също G, се изразяват чрез v, което прави целия проблем нелинейни. При предположението, че адвектираните координати го правят не се деформира твърде силно в рамките на една стъпка, което установихме валидно предположение дори за големите CFL числа, използвани в графиките, ние линеализираме проблема за оптимизация чрез изчисляване на G 0 с помощта на v 0 в уравнение (5). Проблемът с пълната минимизация сега е даден от



където сме въвели мащабиращ коефициент m н , което прави системата независимо от избраната разделителна способност на клетката. Изчисляваме финала позиция на лепенката чрез решаване на линейната система v = (λG 0 + аз) −1v 0 . Имайте предвид, че тази система от уравнения е сравнително малка, с 3м неизвестни на пластир. Следователно, тя може да бъде решена много ефективно с няколко стъпки на конюгатен градиент решаващ, независимо за всички кръпки.

Тъй като нашата цел е да проследяваме широкомащабни движения с нашите пластири, ние трябва да спазват различните пространствени мащаби, обединени в полето на потока. За да намалим ефектите от дребномащабни смущения в потока, ние рекламирайте пачовете с нискочестотна филтрирана версия на скоростта, където филтърът се избира според размера на клетката на клетката ∆xcage.

Очакване. За да предотвратим резки промени в плътността, ние избледняваме кръпки в и извън през интервал от време tf за изобразяване (вижте по-долу). За нашите примери използваме tf от 40 времеви стъпки. За съжаление, това временното избледняване означава, че когато петната са напълно видими, те обикновено вече са силно деформирани поради въртящия се поток движения. Можем да заобиколим този проблем, като оставим кръпките предвидете движението на потока. Т.е., за нов пластир в момент t, ние връщаме движението и деформацията му назад към предишния момент t – tf . Това води до напълно недеформирани конфигурации на кръпка точно при момент във времето, когато са напълно видими.

Инициализация. При засяване на нов, недеформиран пластир в даден момент позиция, открихме, че клетките, подравнени по оста, не е най-доброто опция. Вдъхновени от класическите дескриптори на характеристиките на изображението, ние инициализираме ориентациите на нашите клетки според градиента на полето на плътност. По-конкретно, ние изчисляваме градиентни хистограми в клетката регион Ω и използвайте посоките с най-висок клас като основни посоки. В масивни ъгли на 3D градиентни хистограми могат да бъдат дефинирани с помощта на меридиани и паралели. Разделяме равномерно азимут θi и полярен ъгъл ϕj с размери на стъпката ∆θ = ∆ϕ = π/nb , което води до 2nb и nb подразделения за азимут и полярни ъгъл, съответно. След това 3D векторите се определят като (r, θ,ϕ), където r обозначава радиуса. Единичната сфера е разделена на набор от кошчета {bij }, 0 ≤ i < 2nb , 0 ≤ j < nb , където bij = (1, θi −∆θ/2,ϕj −∆ϕ/2) обозначава нормализираната централна посока на кошчето в сферичната координатна система. Въз основа на тези контейнери се изчисляват хистограмите като:



където o е позицията на точката на пробата в областта Ω с плътност градиент (ro, θo,ϕo ), w(d) = max(0, 1 − d), Aij представлява твърдот ъгъл на bij , а G означава гаусово ядро.

На позиция x на нов пластир, ние изчисляваме градиентната хистограма за плътността на дима ds, както е посочено по-горе. Хистограмата има подразделение от nb = 16, а областта Ω се дефинира като 9 3 мрежа около x. Избираме основната посока на пластира като bk , където k означава хистограмата bin с k = arg maxi,j hijif . За вторичната посока, ние преизчисляваме хистограмата с градиентни вектори в допирателната равнина. По този начин вместо ∇ds използваме (∇ds − (∇ds · bx )bx ). Тогава първоначалната ориентация на пластира е дефинирани по отношение на (bx , by, bx ×by ). Тази инициализация, съобразена с градиента, стеснява потенциалното пространство на дескрипторите, което опростява проблемът с обучението и води до по-стабилни дескриптори.

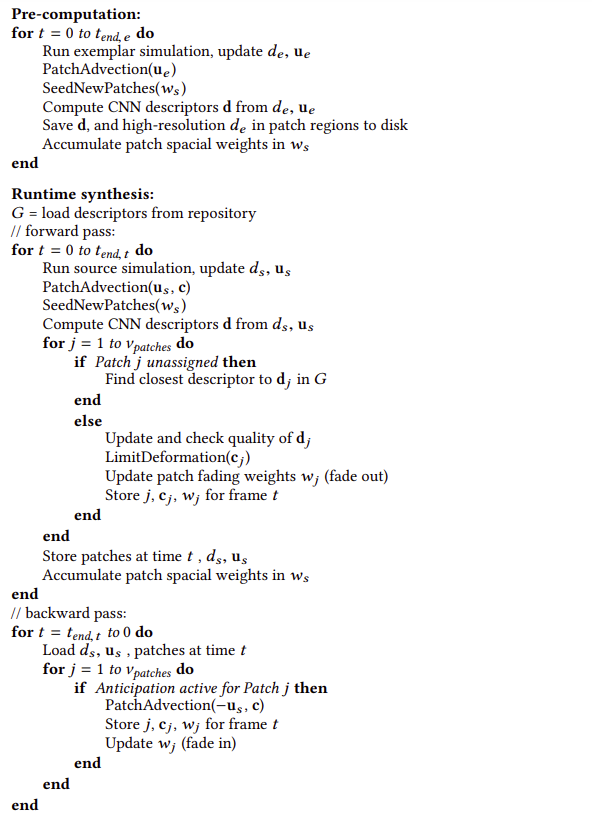
*Дискусия.* Докато възможно най-твърдите деформации имат широко използвани за задача за геометрична обработка, нашите резултати показват, че те също са много полезни за проследяване на течни региони, като същевременно ограничават деформацията. За разлика от типичните задачи за деформация на мрежата, ние нямаме обработва като ограничения, но вместо това допълнителен наказателен срок, който поддържа деформираната конфигурация близка до предписаната от движение на потока. Директното сравнение между обикновена адвекция, и нашите деформации на клетката са показани на фиг. 10. Очевидно е, че директната адвекция е неизползваема на практика, докато нашата деформация ограничаването успешно позволява на клетките да следват потока, като същевременно предотвратяват нежелана деформация

Стъпката на очакване по-горе предизвиква определени разходи за съхранение, но открихме, че значително намалява общата деформация и следователно повишава качеството на синтезираните плътности. Намерихме индуцираните разходи за памет и съхранение да бъдат незначителни на практика

5.2 Синтез и рендерин

По-долу ще очертаем нашия алгоритъм за двупроходен синтез, както и стъпките, необходими за генериране на крайните обеми за изобразяване. Дадено е обобщение на псевдокод на етапа на синтез в Алгоритъм

При предния проход за синтез на потока се поставят нови пластири и отклонени стъпка по стъпка и накрая избледняват. За да поставите нови лепенки, използва се произволна решетка за засяване с разстояние sp/2, като sp = n∆xcage е размерът на пластир. В допълнение, ние използваме претегляне на кръпка мрежа ws . Той използва естествената разделителна способност на симулацията и действа като праг за избягване на засяването на нови петна твърде близо съществуващи такива. ws натрупва пространствените тегла на петна, т.е. сферични зърна, центрирани в центроидите на всяка клетка за лепенки с a радиус на sp/2. Линеен спад се прилага за последните две трети от него радиус, нарастващ от едно до нула. По време на симулация обикновено натрупват теглата на пластира в ws, без да прилагате пластира деформации. Това е в контраст с времето за изобразяване, където се деформираме ядрата на пластира. Решетката за тежести с висока разделителна способност с деформирана теглата на пластира за изобразяване ще бъдат обозначени, за да се разграничат от версията с ниска разделителна способност ws. Тъй като ws се използва само за праг създаването на нови частици, открихме, че използвайки un-deformed kernels дава много добри резултати с намалено време на изпълнение



Нови пачове няма да бъдат въведени на позиция за вземане на проби xn освен ако ws (xn) < 0,5. На практика това означава разстоянието до най-близкият пластир е по-голям от sp/3. За всеки новоприсвоен пластир, ние изчисляваме нейната първоначална референтна рамка, подравнена с градиент уравнение (10), изчислете входовете на CNN на това място и оставете и двете CNN генерира дескрипторите на характеристиките. Въз основа на дескрипторите, ние търсим най-близките съвпадения от нашето хранилище с предварително изчислено kd-дърво. За успешно съчетани кръпки, нашата деформация ограничава адвекцията се извършва през целия им живот. Максималният живот е определя се от дължините на набора от данни на съвпадащите корекции на хранилището, които обикновено са около 100 кадъра. Премахваме неподходящи пластири, чието преоценено разстояние на дескриптора е твърде голямо за текущото конфигурация на потока, или чиято енергия на деформация в уравнение. (7) надвишава праг. На практика открихме праг от 0,15s 2 стр да работиш добре за всички наши примери. След паса напред, съвпадащи кръпки предвидете движението на симулацията на целта при преминаване назад. Тук се движим назад във времето и рекламираме всички новосъздадени петна назад в хода на избледняване в интервал. И накрая, за всеки кадър съхраняваме грубите симулационни плътности, както и пач позиции на върхове v, заедно с техните идентификатори на хранилище и скалар, временни избледняващи тежести. Тези данни са всичко, което е необходимо за етап на изобразяване.

По време на синтеза адвекцията на петна, ограничаваща деформацията, ефективно води до големите движения, които съответстват на входния поток, докато движението в малък мащаб автоматично се извлича от хранилището кадър по кадър, когато изобразяваме изображение. Имайте предвид, че ние само работите с дескрипторите на характеристики с ниски размери, когато синтезирате нов резултат от симулацията, нито една от данните с висока разделителна способност не е необходими на този етап.

По време на рендеринг ние синтезираме крайния обем с висока разделителна способност. За да подготвим този обем, трябва да вземем предвид и пространствените преходи между петна помежду си и преходи от кръпка на данните към оригиналната симулация. За това отново натрупваме деформираните сферични ядра на кръпка в нова мрежа, написана с окончателна разделителна способност на изобразяване, за пространствено претегляне на данните за плътността.

След това картографираме натрупването на претегленото съдържание с висока разделителна способност на набор от данни за корекция в обема за изобразяване с висока разделителна способност. Като нашето хранилище съдържа плътности, които са нормализирани до [0..1], и нашият дескриптор е инвариантен към трансформациите за мащабиране и изместване, ние съпоставете съдържанието на хранилището с мин.-макс. диапазон на плътностите в целевия регион. За да смесите приноса на припокриващите се кръпки, ние нормализираме натрупаната плътност с висока разделителна способност чрез wr. Освен това използваме замъглена версия на оригиналните груби плътности като маска. Забелязахме, че кръпките понякога могат да излязат навън основният обем, докато се движите с потока. Маската за плътност ефективно предотвратява отделянето на плътност на петна оригиналния обем.

6 РЕЗУЛТАТИ И ДИСКУСИЯ

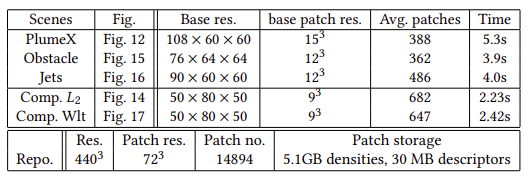


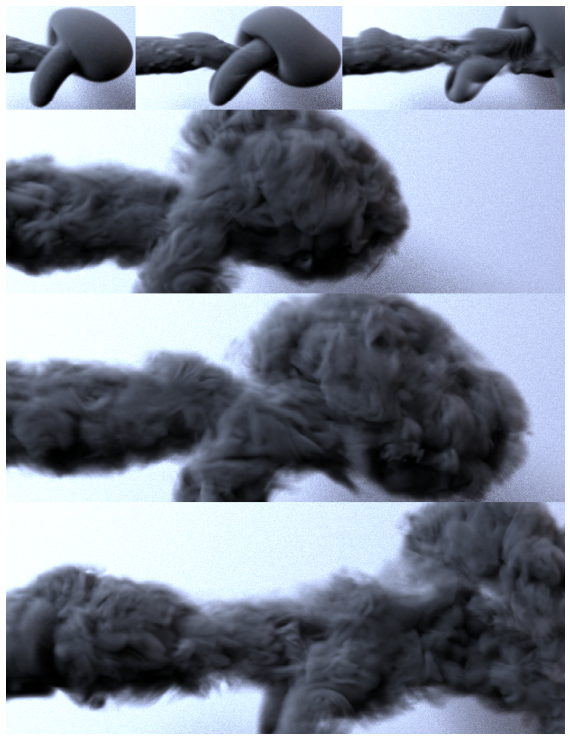
Таблица 1. Подробности за нашите настройки за анимация и генериране на данни от хранилището.

Сега ще демонстрираме, че нашият подход може ефективно да синтезира обеми с висока разделителна способност за различни потоци. Всички времена на изпълнение по-долу са дадени средни стойности на кадър. обикновено, изпълняваме една стъпка във времето на кадър от анимация.

Като първи пример симулирахме прост пример за издигащ се шлейф, показан на фиг. 12. Имайте предвид, че гравитацията действа по оста x в нашата настройка. В този случай разделителната способност на оригиналната симулация е 108 × 60 × 60 и средно 388 пластира са били активни през хода на симулацията. Нашият подход е в състояние да синтезира a голямо количество ясно дефинирани детайли за входния поток, който прави не се разсейват с течение на времето. Подробности за тази настройка на симулацията, както и за другите примери можете да намерите в Таблица 1.

Втори пример е показан на фиг. 15. Тук добавяме допълнително препятствие, което отклонява потока. За движението на пластира, ние просто удължете скоростите на потока в препятствията и добавете лека корекция по наклона на функцията за разстояние със знака на препятствието, ако a върхът на patch неволно се озовава вътре в препятствието. Нашата адвекция, ограничаваща деформацията, плавно насочва петната около регион на препятствия.

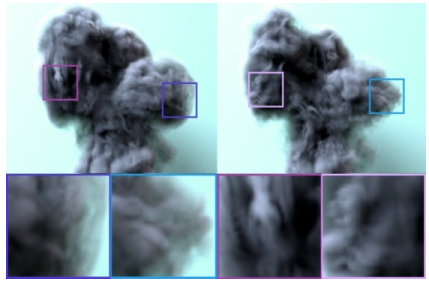
Различна конфигурация на потока със сблъскващи се струи дим е показано на фиг. 16. За тази настройка средно са били активни 486 кръпки. Имайте предвид, че нашите пачове съдържат 723 клетки в този случай. По този начин, на ефективната разделителна способност за тази симулация е около 560 × 360 × 360 клетки.



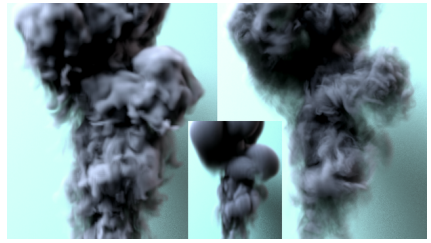
Фиг. 12. Прилагаме нашия метод към симулация на хоризонтален поток. Върха три изображения показват входа, долните три плътностите с висока разделителна способност, генерирани с нашия метод. Средно 388 пластира са били активни на времева стъпка.

Цялата симулация отне само 4.0s/кадър, което е много ефективно предвид високата ефективна разделителна способност. За трите примера по-горе ние използваме ws, натрупани от деформирани ядра на пач. Като се вземе предвид деформацията, ws е по-добре представлява покритието на пластирите. Въпреки това, тъй като ние ограничаваме деформации, открихме, че използването на недеформирани ядра генерира еквивалентни визуални резултати с намалено време за изчисление. Освен това, все още остава значително място за по-нататъшно ускоряване на нашето прилагане. Например, ние стартираме флуидното решение на процесора и само ние използвайте графични процесори за изчисления на дескрипторите на CNN.

Оценки: Дескрипторите на CNN не само увеличават процента на припомняне, но също така подобри визуалното качество чрез извличане на кръпки, които са по-добри се придържат към входа ow. Това може да се види на фиг. 13, където нашият резултат е показан отдясно, докато лявата страна използва нашия пълен конвейер с опростено изчисляване на разстоянието, т.е. без използване на CNN. Вместо това, ние използваме разстояние L2 на понижени извадки версии на ds и къдрицата на нас . Тези стойности се нормализират и се използват като дескриптори директно. По-конкретно, ние намалихме семплирането на ds до разделителна способност от 7 3 , и къдрите до 5 3 , което води до комбиниран дескриптор с (343 + 375) записи. Това е подобно на размера на нашите базирани на CNN дескриптори. Тъй като дескрипторите на CNN имат добро разбиране за корелацията между различни разделителни способности на течности, те предлагат резултати с малки вихри и ярки структури, които пасват добре на целта, докато простите дескриптори понякога предлагат обикновени и шумни структури ( сини области на фиг. 13). Освен това, простите дескриптори могат въвеждат неправдоподобни движения, което става очевидно в региони, отбелязани с лилаво на фиг. 13. Там знаем от теорията че бароклинният принос към завихрянето трябва да бъде по протежение на кръстосано произведение на градиента на плътност и налягане. По този начин вертикалата структурите, причинени от простите дескриптори, не са правдоподобни за симулация на вход, задвижван от плаваемост



Фиг. 13. Използването на нашия алгоритъм с прости дескриптори (вляво) може да доведе до прекалено правилни структури (сини правоъгълници) и неоптимални съвпадения (лилаво правоъгълници). Шансовете за такива присвоявания на корекции са намалени с базираните на CNN дескриптори (вдясно).



Фиг. 14. Въз основа на симулация с 50 × 80 × 50 клетки (център), вълната метод на турбулентност (вляво) генерира обеми със 150 × 240 × 150 клетки с 2,75s/кадър. Нашият метод (вдясно) дава ефективни разделителни способности от 400 × 640 × 400 с 2.23s/кадър.

Въз основа на настройката от фиг. 13, ние допълнително сравняваме нашия подход към метода на вълновата турбулентност като представител от класа на up-res методи. За да се направи подходите са сравними, ние считаме тяхното представяне като а ограничен и еквивалентен изчислителен бюджет. С този бюджет, Wavelet turbulence метод дава резултати с 3 пъти по-високи резолюция, консумираща 2,75 s/кадър. Това е близо до 2.23s/кадър нашият метод изисква. Въпреки това, нашият метод ефективно дава 8 пъти по-висока разделителна способност (виж Таблица 1). За нашата симулация, тази настройка използва полето ws с недеформирани оценки на ядрото. Освен от разликата в детайлите, версията на вълновата турбулентност показва aзабележимо отклонение от входния поток в долната част на сила на звука. Тук числената дифузия се натрупва, за да предизвика значителна се отклоняват във времето, докато нашият метод продължава да се съобразява с него входа.

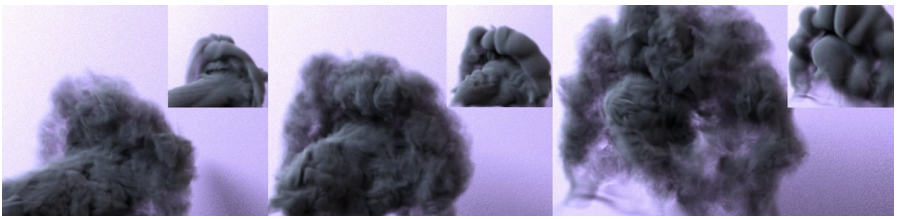
Накрая сравняваме нашия метод с обикновена симулация с удвоена резолюция. Както се очакваше, тази версия води до различно широкомащабно движение и за да сравним изходите, ние приложихменашия метод за синтез, базиран на CNN, към намалена извадка версия на симулация с висока разделителна способност. Докато обикновената висока разделителна способност сцената харчи 2.5s/кадър, нашият метод отнема само 2.42s, но предлага добре подробности, както е показано на Ограничения и дискусия: Едно от ограниченията на нашия подход е това, è не можем да гарантираме напълно свободни от дивергенция движения в малки мащаби.

За по-големи мащаби нашите изходи съответстват на оригиналното движение без отклонения. Дребномащабните движения, съдържащи се в кръпките на хранилището също се записват от потоци, напълно свободни от дивергенция, но като нашите петна се деформират леко, резултантните движения не са гарантирани да бъде без отклонения. Освен това може да се въведе пространствено смесване области с различни движения. Нашият алгоритъм споделя това поведение с други подходи за синтез, например, синтез на текстура. Въпреки това, тъй като не е необходимо да изчисляваме стъпка на адвекция въз основа на тях движения, нашият метод избягва натрупването на загуби (или печалби) на маса с течение на времето. фиг. 17.

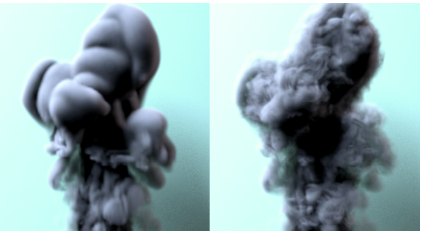
Има много пътища за по-малки подобрения на нашата нервна система мрежов подход, например прилагане на техники като нормализиране на партиди или специализирани техники за конструиране на набора за обучение. Въпреки това, ние вярваме, че настоящият ни подход вече демонстрира, че деформиращите лагранжеви области на петна са отлични основа за CNN във връзка с флуидни потоци. Ефективно прави нашият подход на обучение е инвариантен към движенията в голям мащаб. Премахването на тези инварианти за проблеми с машинното обучение е важно тема, както беше споменато напр. от Ladicky et al. Освен движението инвариантност, стигаме до алгоритъм, който лесно може да се приложи към всеки разделителна способност на източника. За други базирани на CNN подходи това е типично много трудно постижимо, тъй като мрежите са специално обучени за a фиксиран входен и изходен размер.



Фиг. 15. Нашият метод, приложен към потока около цилиндрично препятствие. Разделителната способност на основната симулация е само 76 × 64 × 64.



Фиг. 16. Две сблъскващи се струи дим, симулирани с нашия подход. Цялата симулация, включително изчисления на дескрипторите, търсене и адвекция на кръпки, взеха средно само 4,0 секунди на кадър.



Фиг. 17. За да сравните пълна симулация от 100 × 160 × 100 (вляво) с нашия подход намаляваме пълната симулация до 50 × 80 × 50 и след това приложете нашия алгоритъм (вдясно). Последният харчи 2.42s/кадър, докато пълният симулацията изисква 2.51s/кадър.

Поради своето естество, управлявано от данни, нашият метод изисква повече твърд диск пространство, отколкото процедурни методи. Както е показано в таблица 1, плътността данните за кръпките в нашето 3D хранилище заемат около. 5,1 GB от дисково пространство. За щастие, ние зареждаме дескрипторите само при симулация време: ок. 15MB за дескриптори на плътност и още 15MB за curl дескриптори. По време на рендиране трябва да заредим ca. 400 набора от данни на рамка, т.е. ок. 137MB общо.

Друго следствие от нашия подход за машинно обучение е това нашата мрежа е специално адаптирана към набор от алгоритмични избори. По този начин, за различен решаващ, ще бъде изгодно да се обучи отново мрежа с подходящ набор от данни и адаптиран интервал tr . Ще бъде интересна област на бъдеща работа дали достатъчно дълбока CNN може да се научи да изчислява дескриптори за по-големи класове решаващи задачи.

7 ЗАКЛЮЧЕНИЯ

Представихме нов метод, базиран на CNN, за реализиране на практически управляван от данни работен поток за симулации на дим. Нашият подход е първият, който използва поточно хранилище от пространствено-времеви набори от данни за синтезиране резултати с висока разделителна способност само с няколко секунди време за изпълнение на кадър. В същото време нашата работа представлява първа демонстрация на полезността на ученето на дескриптори в контекста на флуидни потоци и ние показахме, че ни позволява да установим съответствия между различни симулации при наличие на числен вискозитет. Като нашите подходът е базиран на данни, може да се използва за всеки избор на NavierStokes решаващ, стига да има достатъчно входни данни. Освен това, локализираните дескриптори правят нашия подход независим от разделителна способност на симулацията

Вярваме, че посоката на синтеза на потока, управляван от данни, е много обещаваща област за приложения за компютърна графика. Арт-директно Решателите, които са в същото време бързи и стабилни, са много търсени. В бъдеще вярваме, че ще бъде много интересно да разширим нашите идеи към стилизиране на потоците, а и към синтезиране не само адвектирано количество като плътност на дима, но скоростта на потока себе си.