

ردیابی حرکت به روش مش وفقی

نسترن برجیان^{*}، سعید صدری[†]، رسول امیرفتاحی[‡]

چکیده

ردیابی حرکت با روش مش دو بعدی بر مبنای ایجاد یک شبکه مش بر روی تصویر و اجازه تغییر شکل به هر سلول مش در فاصله هر دو فریم متوالی با الزام به پیوستگی مش می‌باشد. بنابراین این روش بسیاری از اغتشاشات روش بلوکینگ که در جبران حرکت و یا ردیابی حرکت با روش تطبیق بلوک معمول می‌باشند را به طور مؤثری حذف می‌کند. در مدل‌های مش یکنواخت، الزام به پیوستگی ساختار مش در کل فریم تصویر می‌باشد، در صورتی که واضح است چنین چیزی در مرزهای همپوشانی اشیاء مناسب نیست. برای فائق آمدن بر چنین محدودیتی، روش مش وفقی که در آن الگوریتم‌های شناسایی نواحی ای از پس زمینه که در فریم بعدی پوشانده خواهند شد (background to be covered-BTBC) و نواحی ای که انجام جبران حرکت در مورد آنها نشان دهنده عدم پیوستگی در حرکت است (model failure-MF) بکار گرفته می‌شود مورد استفاده قرار می‌گیرد. در مدل مش یکنواخت، یک مجموعه مشخص از نقاط که در واقع نقاط گوشه سلول‌های ساختار مش هستند در کلیه فریم‌های دنباله تصاویر ویدئویی دنبال می‌شوند و واضح است که در نواحی ای که اشیاء با یکدیگر همپوشانی کرده اند و یا بعضی اشیاء در حال خروج از صحنه تصویر و بعضی دیگر در حال ورود به صحنه تصویر هستند، این روش کارایی خود را از دست می‌دهد. در حالی که در روش مش وفقی، در هر فریم با توجه به نواحی ای که در بالا ذکر شد مجدداً مجموعه نقاط و به تبع ساختار مش باز تعریف می‌شود. در روش مش وفقی، پیوستگی ساختار مش در نواحی پوشیده شده و پوشیده نشده شکسته می‌شود. این امر بوسیله عدم قرار دهی نقطه ای در پس زمینه پوشیده شده و باز تعریف ساختار مش در ناحیه MF در هر فریم محقق می‌گردد. ما الگوریتم طراحی مش بر پایه محتوای هر سلول و انتخاب نقاط برای طراحی مش مثلثی دو بعدی و ردیابی حرکت با روش مش وفقی در حضور همپوشانی ارائه شده توسط Yucel Altunbasak را بهبود داده و سپس یک روش مثلث سازی جدید و همچنین یک الگوریتم جدید برای اطمینان از پیوستگی ساختار مش پس از تخمین بردار حرکت نقاط مش را ارائه می‌کنیم. از آنجا که این ساختار مش بهبود یافته به صورت وفقی بوده و بر مشخصات هر سلول استوار می‌باشد، لذا دیگر لزومی به انتقال کلیه نقاط در هر فریم تصویر وجود ندارد.

کلمات کلیدی

ردیابی، مش وفقی، تخمین حرکت، تقریب polygon، مثلث سازی، جبران حرکت، باز تعریف مش.

Adaptive forward-tracking mesh model

Nastaran Borjia, Saied Sadri, Rassoul Amirfatahi

Abstract

Two-dimensional mesh-based motion tracking preserves neighboring relations (through connectivity of the mesh) and also allows warping transformations between pairs of frames; thus, it effectively eliminates blocking artifacts that are common in motion compensation by block matching. However, available uniform 2-D mesh model enforces connectivity everywhere within a frame, which is clearly not suitable across occlusion boundaries. To overcome this limitation, BTBC (background to be covered) detection and MF (model failure) detection algorithms are being used. In this algorithm, connectivity of the mesh elements (patches) across covered and uncovered region boundaries are broken. This is achieved by allowing no node points within the background to be covered and refining the mesh structure within the model failure region at each frame. We modify the occlusion-adaptive, content-based mesh design and forward tracking algorithm used by Yucel Altunbasak for selection of points for triangular 2-D mesh design. Then, we propose a new triangulation procedure for mesh structure and also a new algorithm to justify connectivity of mesh structure after motion vector estimation of the mesh points. The modified content-based mesh is adaptive which eliminates the necessity of transmission of all node locations at each frame.

Keywords

Tracking, adaptive mesh, motion estimation, polygon approximation, triangulation, motion compensation, refinement of the mesh.

^{*} دانشگاه صنعتی اصفهان، دانشکده مهندسی برق و کامپیوتر، n_borjia2002@yahoo.com

[†] عضو هیات علمی دانشگاه صنعتی اصفهان، دانشکده مهندسی برق و کامپیوتر، sadri@cc.iut.ac.ir

[‡] عضو هیات علمی دانشگاه صنعتی اصفهان، دانشکده مهندسی برق و کامپیوتر، fattahi@cc.iut.ac.ir

۱- مقدمه

مدل مش دو بعدی یک روش ردیابی حرکت است که بر مبنای تغییر شکلهای فضایی در سلولهای ساختار مش و نسبتهای همسایگی بین سلولهای مختلف می‌باشد. حرکت در این روش از طریق انتقال نقاط گوشه سلولهای ساختار مش ردیابی می‌شود که این انتقالها با استفاده از میدان حرکت تخمینی محاسبه می‌گردند [۲،۱]. مشهای دو بعدی را می‌توان به مشهای یکنواخت با یک اندازه یکسان برای همه سلولها [۳] و مشهای غیر یکنواخت مانند مشهای hierarchical و content-based که اندازه سلولها در آنها با توجه به صحنه تصویر به صورت وقتی تغییر می‌کند تقسیم کرد [۶،۵،۴].

سلولها در مش یکنواخت ممکن است در فریمهای متوالی دنباله تصاویر ویدئویی با یکدیگر همپوشانی داشته باشند. در حالی که یک مش غیر یکنواخت بر روی مرزهای اشیاء وفق داده می‌شود. علاوه بر این زمانی که بیش از یک نوع حرکت در صحنه تصویر وجود دارد، مش یکنواخت از کارایی کافی برخوردار نیست [۸،۷].

در حقیقت، ردیابی شیء، مشخص کردن تغییرات موقعیت مکانی شیء و دنبال کردن آن در یک دنباله تصاویر ویدئویی با یک هدف خاص می‌باشد. یک ردیابی موفقیت آمیز به میزان زیادی به نحوه و دقت شناسایی نواحی همپوشانی و نواحی MF و تخمین میدان حرکت در نزدیکی مرزهای اشیاء بستگی دارد. جبران حرکت نیز برای تعیین نقاطی که باید از مجموعه نقاط ساختار مش حذف گردند و مشخص کردن مکان نقاط جدید مورد نیاز است.

در این مقاله، ما یک پروسه ردیابی مش وقتی را بیان می‌کنیم. در این روش، به هیچ نقطه ای اجازه قرار گیری در نواحی ای از پس زمینه که در فریم بعدی پوشیده خواهد شد (نواحی BTBC) داده نخواهد شد و ساختار مش در نواحی MF برای ردیابی متوالی این نواحی باز تعریف خواهد شد [۹].

این مقاله شامل بخشهای زیر است: در بخش دوم، ما نخست اصول پایه الگوریتم مش وقتی، تخمین میدان حرکت و شناسایی ناحیه BTBC را بیان می‌کنیم. سپس تقریب polygon بررسی می‌شود. پس از آن، ما یک پروسه جدید و کارآمد طراحی مش وقتی که برای طراحی مش ابتدایی بکار می‌رود را بیان می‌کنیم. در ادامه، ما یک روش جدید مثلث سازی را ارائه و پیاده سازی می‌کنیم. بخش سوم شامل تخمین حرکت هر نقطه است که در قسمتهای جبران حرکت و ردیابی مش مورد استفاده قرار می‌گیرد. در ادامه نیز یک الگوریتم باز تعریف و ردیابی به سمت جلوی مش بیان می‌شود. نتایج تجربی روی تصاویر ساختگی و تصاویر ساده واقعی نشان دهنده موفقیت الگوریتم ردیابی مش وقتی بیان شده در مقایسه با مش یکنواخت برای ردیابی شیء و در حضور همپوشانی است.

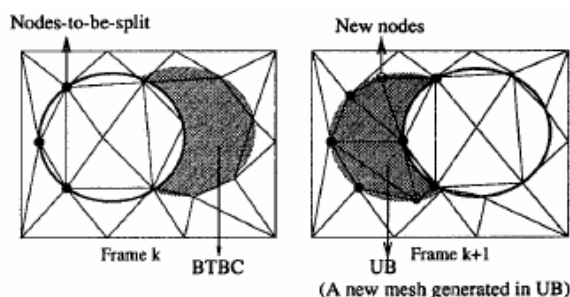
۲- مش وقتی

در این قسمت به اصول پایه الگوریتم مش وقتی پرداخته می‌شود.

۲-۱- مفهوم مش وقتی

در مدلهای مش استاندارد، الزام به پیوستگی حرکت در کل فریمها وجود دارد و به روشنی چنین چیزی در مواقعی که حرکتها چندگانه و نواحی همپوشانی در صحنه تصویر وجود دارد مطلوب نیست [۲،۱]. مفهوم مش وقتی نیز برای فائق آمدن بر این محدودیت مهم در ردیابی حرکت بیان شد.

به عنوان مثال، یک شیء بیضی شکل متحرک را مطابق شکل ۱ در نظر بگیرید.



شکل ۱: مفهوم مش وقتی

فرض کنید که شیء در حال حرکت به سمت راست است؛ بنابراین دو نوع حرکت در صحنه تصویر وجود خواهد داشت: حرکت به سمت راست که به ناحیه BTBC تخصیص داده می‌شود و حرکت به سمت چپ که به ناحیه ای از پس زمینه تخصیص داده می‌شود که در فریم جاری پوشیده شده است و در فریم بعدی دیگر پوشیده نخواهد بود و به آن ناحیه uncovered background-UB گفته می‌شود. توجه کنید که ناحیه BTBC در فریم k باید کاملاً در فریم k+1 پوشیده شود و ناحیه UB در فریم k+1 نیز باید کاملاً در فریم k پوشیده باشد.

۲-۲- تخمین میدان حرکت

الگوریتم تخمین میدان حرکت استفاده شده در این مقاله، بردارهای حرکت را به طور مستقل در هر پیکسل از یک میدان شار نوری یا optical flow با استفاده از روش Horn-Schunck [۱۰] تخمین می‌زند. یک روش تخمین میدان حرکت بر پایه میدان شار نوری بهتر از روشهای دیگر به عنوان مثال روش تطبیق بلوک است زیرا روشهای بر پایه شار نوری میدانهای حرکت یکنواخت تری که برای پارامتری کردن میدان حرکت مناسب تر است ایجاد می‌کنند [۱۱]. در [۱۲] روش دیگری نیز برای تخمین میدان حرکت بر پایه روش split-and-merge توسط Lucas و Kanade بیان شده است.

۳-۲- شناسایی ناحیه BTBC

الگوریتم شناسایی ناحیه BTBC که در [۲،۱] مورد استفاده قرار گرفته است شامل مراحل زیر می‌باشد:

۱. محاسبه اختلاف فریم دو فریم متوالی و تعیین ناحیه تغییر یافته با استفاده از آستانه گذاری و اعمال پردازشی به صورت زیر:

الف) فیلتر کردن با استفاده از فیلتر median با اندازه 5×5 .

ب) سه مرتبه استفاده متوالی از عمل مورفولوژی closing با اندازه 3×3 و سپس سه مرتبه استفاده متوالی از عمل مورفولوژی opening با همان اندازه.

ج) حذف نواحی کوچک که کوچکتر از یک اندازه از پیش تعیین شده برای مثال ۲۵ پیکسل می‌باشند.

۲. تخمین میدان حرکت از فریم k به فریم $k+1$.

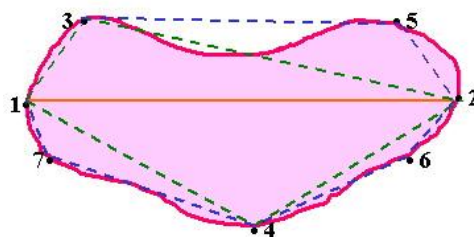
۳. جبران حرکت با استفاده از میدان حرکت برای فریم k از فریم $k+1$ برای محاسبه فریم مجازی $\tilde{I}_k$.

۴. اگر اختلاف بین $I_k(x, y)$ و $\tilde{I}_k(x, y)$ بزرگتر از یک آستانه از پیش تعیین شده است، پیکسل (x, y) به عنوان یک پیکسل همپوشانی نشانه گذاری می‌شود. به عبارت دیگر:

$$(1) \quad \text{همپوشانی شده } (x, y) \Rightarrow T_1 > |I_k(x, y) - \tilde{I}_k(x, y)| \text{ اگر}$$

۴-۲- تقریب polygon

زمانی که لازم است مرز یک ناحیه بوسیله یک مجموعه نقاط تقریب زده شود به گونه ای که این مجموعه نقاط باید نزدیکترین تقریب به ناحیه مورد نظر با تعداد محدودی پارامتر را فراهم کند، تقریب polygon می‌تواند بکار برده شود. این تقریب همانگونه که در شکل ۲ مشاهده می‌شود به طور طبیعی بر مرزهای یک ناحیه منطبق می‌شود.



شکل ۲: تقریب polygon

الگوریتم تقریب polygon دارای مراحل زیر است [۱۳،۱]:

۱. دو پیکسل (۱ و ۲) روی مرز ناحیه که دارای ماکزیمم فاصله از یکدیگر می‌باشند را بیابید. خط واصل بین این دو نقطه، محور اصلی نامیده می‌شود.

۲. دو نقطه (۳ و ۴) روی مرز در دو جهت محور اصلی که دارای بزرگترین فاصله عمودی از محور اصلی هستند را بیابید. این چهار نقطه را نقاط اصلی می‌نامند.

۳. اکنون، چهار قسمت ۱-۳، ۲-۳، ۳-۴ و ۴-۱ وجود دارد. قسمت ۱-۳ را در نظر بگیرید. یک خط مستقیم از نقطه ۱ به نقطه ۳ ترسیم کنید. هر پیکسل روی مرز که بین ۱ و ۳ قرار دارد و هر پیکسل که در ناحیه بین مرز و خط مستقیم قرار دارد را شناسایی کنید. اگر ماکزیمم فاصله عمودی بین خط مستقیم و هر پیکسل روی مرز کمتر از یک آستانه خاص است و پیکسلهایی که در ناحیه بین مرز و خط مستقیم قرار دارند کمتر از مثلاً ۵٪ کل پیکسلهای ناحیه مورد نظر هستند، آنگاه لازم نیست هیچ نقطه جدیدی در این قسمت از مرز قرار داده شود. اما اگر هر دو شرط فوق با هم محقق نمی‌شوند، باید یک نقطه جدید روی مرز در محل پیکسل با ماکزیمم فاصله از خط مستقیم قرار داده شود. این پروسه ادامه می‌یابد تا وقتی که هیچ نقطه جدیدی لازم نباشد در قسمتهای مرزی که جدیداً ایجاد شده اند قرار داده شود.

۴. مرحله ۳ برای سه قسمت مرزی باقیمانده تکرار می‌شود.

نتایج پیاده سازی الگوریتم فوق برای چندین ناحیه دلخواه در شکل ۳ نشان داده شده است. معیار توقف الگوریتم را می‌توان تعداد ماکزیمم نقاطی که الگوریتم مجاز به قرار دهی روی مرز است قرار داد. اگر تعداد نقاط قرار داده شده روی مرز بیشتر از این تعداد ماکزیمم برای مثال ۶۴ نقطه شود، الگوریتم متوقف می‌شود؛ اگر چه در عمل معمولاً این تعداد نقطه لازم نمی‌شود. الگوریتم پیاده سازی شده در اینجا دارای معیار توقف ۶۴ نقطه است ولی به آسانی قابل بسط به ۱۲۸ یا ۲۵۶ نقطه می‌باشد. از دیگر مزایای الگوریتم پیاده سازی شده، می‌توان به عدم وجود هیچ محدودیت خاصی بر روی شکل و اندازه ناحیه تقریب زده شده اشاره کرد.

پس از انتخاب نقاط، یک پروسه مثلث سازی با استفاده از نقاط انتخاب شده انجام می‌شود. الگوریتم دارای مراحل زیر است:

۱. ناحیه شی را با تقریب polygon تقریب بزنید.
۲. همه پیکسلها به جزء آنها که در ناحیه BTBC قرار دارند را به صورت unmark نشانه گذاری کنید. DFD میانگین را برای همه پیکسلهای unmark با استفاده از رابطه زیر محاسبه کنید:

$$DFD_{avg} = \frac{\sum_{(x,y)} (DFD(x,y))^p}{k} \quad (2)$$

$$DFD(x,y) = |I_k(x,y) - \tilde{I}_k(x,y)| \quad (3)$$

در رابطه ۲، پارامتر k تعداد پیکسلهای unmark است و p می‌تواند مقدار ۱ یا ۲ انتخاب شود.

- ۳ یک دایره حول هر نقطه که از تقریب polygon بدست آمده است تا زمانی که $\sum_{(x,y)} (DFD(x,y))^p$ در این دایره بزرگتر از DFD میانگین است رشد بدهید. همه پیکسلهای درون این دایره را به صورت marked نشانه گذاری کنید.

۴ ناحیه BTBC را نشانه گذاری کنید.

- ۵ یک تابع ارزش برای هر پیکسل unmark با استفاده از رابطه زیر محاسبه کنید:

$$C(x,y) = |I_x(x,y)| + |I_y(x,y)| \quad (4)$$

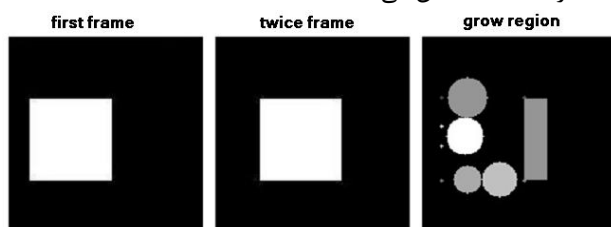
که $I_x(x,y)$ و $I_y(x,y)$ مشتقات جزئی هستند.

- ۶ یک پیکسل unmark که دارای بزرگترین مقدار تابع ارزش $C(x,y)$ است را انتخاب کنید و یک ناحیه دایره ای شکل با یک شعاع از پیش تعریف شده در حول آن در نظر بگیرید. اگر هیچ نقطه marked دیگری در این دایره وجود ندارد، این پیکسل جدید را نشانه گذاری کنید.

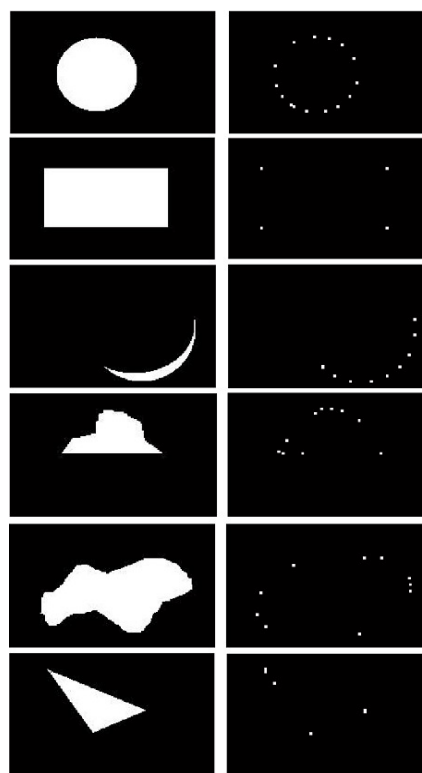
- ۷ یک ناحیه دایره ای شکل در حول این پیکسل جدید مشابه مرحله ۳ رشد بدهید. زمانی که ناحیه رشد داده شده در حال برخورد با یک ناحیه marked دیگری است، رشد دایره را متوقف کنید حتی اگر شرایط مرحله ۳ محقق نشده باشد.

- ۸ مرحله ۶ را تا زمانی که یک تعداد مطلوب نقطه انتخاب شده باشد تکرار کنید.

شکل ۵ نتایج پروسه فوق را برای یک مربع که در حال حرکت به سمت راست است نشان می‌دهد.



شکل ۵: نتایج الگوریتم انتخاب نقاط برای مش دو بعدی

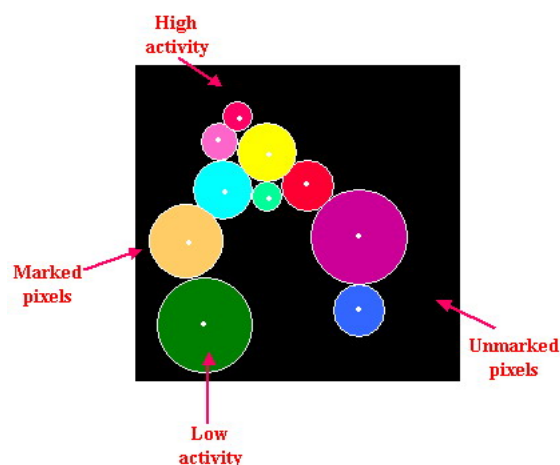


شکل ۳: نتایج پیاده سازی الگوریتم تقریب polygon

۲-۵- انتخاب نقاط برای طراحی مش دو بعدی مثلثی

در این قسمت، ما یک الگوریتم جدید و مؤثر انتخاب نقاط را برای طراحی مش دو بعدی مثلثی بیان می‌کنیم.

این الگوریتم یک مش غیر یکنواخت را ایجاد می‌کند به گونه ای که چگالی نقاط متناسب با شدت تغییرات محلی حرکت است و مرزهای مش منطبق بر مرزهای اشیاء می‌شود. بنابراین، نقاط باید در روی لبه ها و یا در محل پیکسلهای با بیشترین گرادیان قرار داده شوند. این پروسه در شکل ۴ نشان داده شده است. دایره های بزرگ نشان دهنده شدت تغییرات کم در اطراف نقطه مورد نظر و دایره های کوچک نشان دهنده شدت تغییرات زیاد در اطراف آن نقطه هستند.



شکل ۴: انتخاب نقاط برای طراحی مش دو بعدی

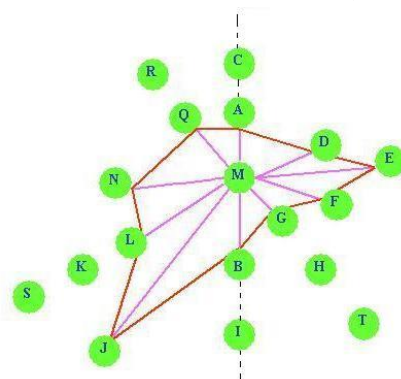
۲-۶- مثلث سازی

ما در این قسمت یک الگوریتم سریع و آسان برای مثلث سازی با نام الگوریتم star-polygon را ابداع کرده و ارائه می‌کنیم. این الگوریتم شامل مراحل زیر است:

- ۱ نقطه M از مجموعه نقاط مش را در نظر بگیرید.
- ۲ نزدیکترین نقطه همسایه M که درست در بالای آن قرار دارد را پیدا کنید و در جدول همسایگی M قرار دهید.
- ۳ نزدیکترین نقطه همسایه M که درست در پایین آن قرار دارد را پیدا کنید و در جدول همسایگی M قرار دهید. شکل ۶ را ببینید.
- ۴ صفحه تصویر را بوسیله یک خط عمودی گذرا از نقطه M به دو قسمت تقسیم کنید.
- ۵ همه نقاط نیم صفحه راست را به عنوان همسایگان نقطه M در نظر بگیرید. اگر دو نقطه روی یک خط یکسان گذرا از نقطه M قرار دارند، نقطه ای که به M نزدیکتر است را نگه دارید و نقطه دیگر را از مجموعه همسایگان حذف کنید. این مرحله را آنقدر تکرار کنید تا هیچ دو نقطه ای در مجموعه همسایگان بر روی یک خط یکسان گذرا از نقطه M قرار نداشته باشند.

- ۶ مرحله ۵ را برای همه نقاط نیم صفحه چپ تکرار کنید.
- ۷ نقطه M را به همسایگانش متصل کنید. این کار باعث ایجاد یک گراف ستاره ای در اطراف نقطه M می‌گردد. واضح است که بر روی هر یک از شعاعهای این ستاره تنها یک نقطه وجود دارد.

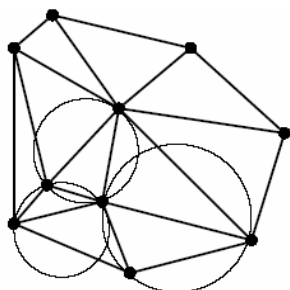
- ۸ همه شعاعهای این ستاره را از بالا به پایین و در جهت عقربه های ساعت جاروب کنید و با هر دو شعاع متوالی یک مثلث بسازید تا وقتی که polygon اطراف نقطه M کامل شود.
- ۹ الگوریتم را برای هر نقطه از مجموعه نقاط مش تکرار کنید.



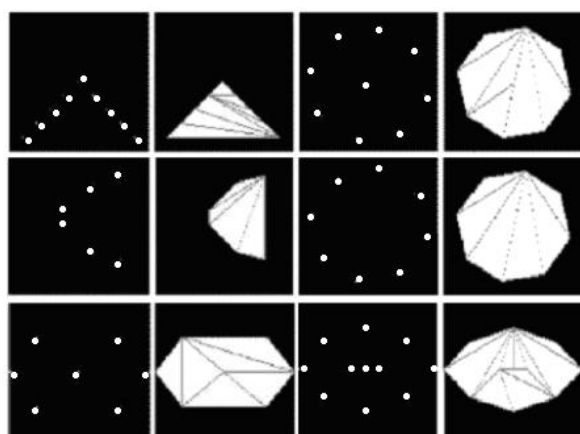
شکل ۶: مثلث سازی star-polygon

مثلثهای ساخته شده نباید با یکدیگر همپوشانی داشته باشند؛ بنابراین، ما از شرط delaunay triangulation برای ساخت مثلثهای بدون همپوشانی استفاده کردیم. delaunay triangulation یک شرط برای

مثلث سازی است و بر اساس آن، اگر یک دایره به یک مثلث ساخته شده محیط شود، هیچ نقطه دیگری نباید در درون این دایره قرار داشته باشد. این شرط در شکل ۷ نشان داده شده است. نتایج الگوریتم مثلث سازی برای چند مجموعه دلخواه از نقاط نیز در شکل ۸ نشان داده شده است.



شکل ۷: delaunay triangulation



شکل ۸: نتایج الگوریتم مثلث سازی

۳- ردیابی حرکت

ردیابی حرکت شامل قسمتهای زیر می‌شود: (۱) تخمین بردار حرکت برای نقاط انتخاب شده (۲) جبران حرکت affine (۳) شناسایی ناحیه model failure (۴) باز تعریف ساختار مش.

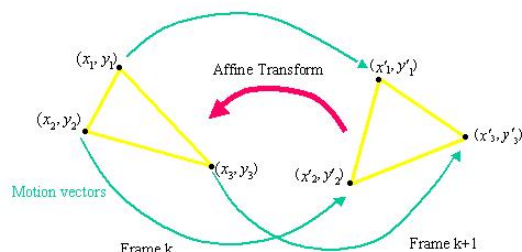
۳-۱- تخمین بردار حرکت برای نقاط انتخاب شده

تخمین بردار حرکت برای نقاط انتخاب شده می‌تواند با استفاده از میدان حرکت محاسبه شده در قسمت شناسایی ناحیه BTBC انجام شود. اما ممکن است بردارهای حرکت تخمینی به گونه ای باشند که پیوستگی ساختار مش را تضمین نکنند. این امر در شکل ۹ نشان داده شده است.

انجام مراحل زیر موجب تضمین پیوستگی ساختار مش می‌شود.

- ۱ نقطه N_0 و همه نقاط متصل به آن را در نظر بگیرید.
- ۲ نقاط متناظر این نقاط از فریم جاری به فریم بعدی را پیدا کنید و آنها را N'_i بنامید.
- ۳ نقطه مجازی N'_0 را با استفاده از رابطه زیر محاسبه کنید:

- ۲ همه نقاط درون این سلول را با استفاده از پارامترهای محاسبه شده و رابطه ۶ جبران حرکت کنید.
- ۳ مرحله ۱ و ۲ را برای همه سلولهای مثلثی تکرار کرده و فریم مجازی $\tilde{I}_{k+1}$ را بدست آورید.



شکل ۱۰: جبران حرکت affine

۳-۳- شناسایی ناحیه model failure

ناحیه model failure با آستانه گذاری اختلاف بین فریم واقعی I_{k+1} و فریم مجازی $\tilde{I}_{k+1}$ بدست می‌آید. به عبارت دیگر، باید رابطه زیر محقق گردد.

$$|\tilde{I}_{k+1} - I_{k+1}| < T_2 \quad (8)$$

۳-۴- باز تعریف ساختار مش

ساختار مش با استفاده از همه نقاط (x', y') باز تعریف می‌شود. ولی اگر یک نقطه در ناحیه MF یا ناحیه BTBC بین فریم $k+1$ و $k+2$ قرار داشته باشد، به دلیل اینکه برای چنین نقطه‌ای نمی‌توان یک بردار حرکت صحیح تعریف کرد؛ آن نقطه از مجموعه نقاط ساختار مش حذف می‌شود. همچنین یک مجموعه نقاط جدید در ناحیه MF قرار داده می‌شود. ناحیه تشخیص داده شده برای شی به عنوان ناحیه مرجع اولیه برای فریم بعدی بکار می‌رود. نتایج الگوریتم ردیابی حرکت با مش وقتی در شکل ۱۱ نشان داده شده است. در ردیف اول، سه فریم متوالی که در اثر حرکت دوربین ایجاد شده اند نشان داده شده است. ردیف دوم شامل ناحیه BTBC و ناحیه همپوشانی بین فریم اول و دوم و همچنین نتایج انجام تابع ارزش می‌باشد. ردیف سوم ساختار مش، جبران حرکت آن و فریم مجازی $\tilde{I}_{k+1}$ را نشان می‌دهد. ردیف چهارم نیز شامل ناحیه MF، ناحیه BTBC بین فریم دوم و سوم و در نهایت ناحیه پیش بینی شده شی در فریم بعدی می‌باشد.

۴- نتیجه

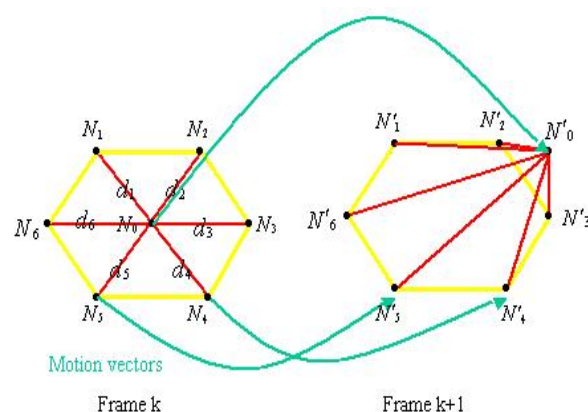
در این مقاله، با توجه به توضیحات داده شده در مورد روش مش وقتی در ردیابی حرکت، با کارایی این روش و همچنین مزایای این روش نسبت به روشهای بلوکینگ و مش یکنواخت آشنا شده و الگوریتمهای ردیابی حرکت مخصوصاً الگوریتم مش وقتی را مورد بررسی قرار داده و معادله optical flow را برای تخمین میدان حرکت و همچنین تقریب

$$u = \frac{\sum_{i=1}^K \frac{u_i}{d_i}}{\sum_{i=1}^K \frac{1}{d_i}} \quad \text{و} \quad v = \frac{\sum_{i=1}^K \frac{v_i}{d_i}}{\sum_{i=1}^K \frac{1}{d_i}} \quad (5)$$

که (u, v) بردار حرکت نقطه مجازی N'_0 و (u_i, v_i) بردار حرکت هر نقطه N_i متصل به N_0 و d_i فاصله بین آنها است.

۴ یک polygon با استفاده از نقطه مجازی N'_0 و دیگر نقاط N'_i بسازید. این کار مشابه روش توضیح داده شده در قسمت مثلث سازی انجام می‌شود.

۵ اگر N'_0 واقعی در درون polygon قرار دارد، این نقطه N'_0 به عنوان نقطه متناظر با N_0 انتخاب می‌شود وگرنه باید نقطه مجازی N'_0 را انتخاب کرد.



شکل ۹: عدم پیوستگی در ساختار مش

۳-۲- جبران حرکت affine

هر پیکسل (x, y) در فریم k به یک پیکسل (x', y') در فریم $k+1$ مطابق شکل ۱۰ متناظر می‌شود. این تناظرها در جبران حرکت affine به صورت رابطه ۶ می‌باشد.

$$\begin{aligned} x &= a_1 x' + a_2 y' + a_3 \\ y &= a_4 x' + a_5 y' + a_6 \end{aligned} \quad (6)$$

متعاقب روابط فوق، الگوریتم جبران حرکت affine دارای مراحل زیر است:

- ۱ پارامترهای affine را برای یک سلول مثلثی با استفاده از رابطه زیر محاسبه کنید:

$$\begin{bmatrix} x'_1 & y'_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x'_2 & y'_2 & 1 \\ x'_3 & y'_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x'_3 & y'_3 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \end{bmatrix} \quad (7)$$

Video. IEEE, transactions on circuits and systems for video technology, VOL. 9, NO. 2. PP.353-369.

[۵] Altunbasak, Y., Al-Regib, G., 2001. 2-D Motion Estimation With Hierarchical Content-Based Meshes. IEEE. PP. 1621-1624.

[۶] Laurent, N., 2000. Hierarchical Mesh-Based Global Motion Estimation, Including Occlusion Areas Detection. IEEE. PP. 620-623.

[۷] Weizhao, J., Wang, P., 2002. An Object Tracking Algorithm Based On Occlusion Mesh Model. In Proc. First International Conference on Machine Learning and Cybernetics, PP. 288-292.

[۸] Toklu, C., 1996. 2-D Mesh-Based Tracking of Dformable Objects With Occlusion. IEEE. PP. 933-936.

[۹] Toklu, C., 1997. 2-D Mesh-Based Synthetic Transfiguration of an Object With Occlusion. IEEE. PP. 2649-2652.

[۱۰] Horn, B. K. P., Schunck, B. G., 1981. Determining Optical Flow, VOL. 17, PP. 185-203.

[۱۱] Tekalp, A. M., 1995. The book, Digital Video Processing. Englewood Cliffs, Prentice-Hall, 1st edition.

[۱۲] Locas, B. D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In Proc. DARPA Image Understanding Workshop, PP. 121-130.

[۱۳] Gerken, P., 1994. Object-Based Analysis-Synthesis Coding of Image Sequences at Very Low Bit-Rates. IEEE Trans. Circuits Syst Video. Technol., VOL. 4, PP. 228-235.

polygon را پیاده سازی کردیم. به علاوه، روش طراحی مش بر مبنای محتوای تصویر را بهبود داده و یک روش جدید برای مثلث سازی و همچنین یک روش برای اطمینان از پیوستگی ساختار مش را ارائه کرده و پروسه های جبران حرکت affine و باز تعریف ساختار مش را پیاده سازی نمودیم. قسمت های بهبود داده شده الگوریتم با دید نظری و نه با مقایسه با نتایج موجود بوده است زیرا با توجه به جدید بودن روش مش و فقی و کمبود مراجع در رابطه با آن و همچنین عدم پیاده سازی کامل الگوریتم بیان شده در این مقاله بوسیله مراجع قابل دسترس، امکان مقایسه نتایج الگوریتم فوق وجود ندارد. در هر صورت، نتایج شبیه سازی حاکی از صحت عملکرد الگوریتم پیاده سازی شده و بخش های پیشنهادی می باشد.

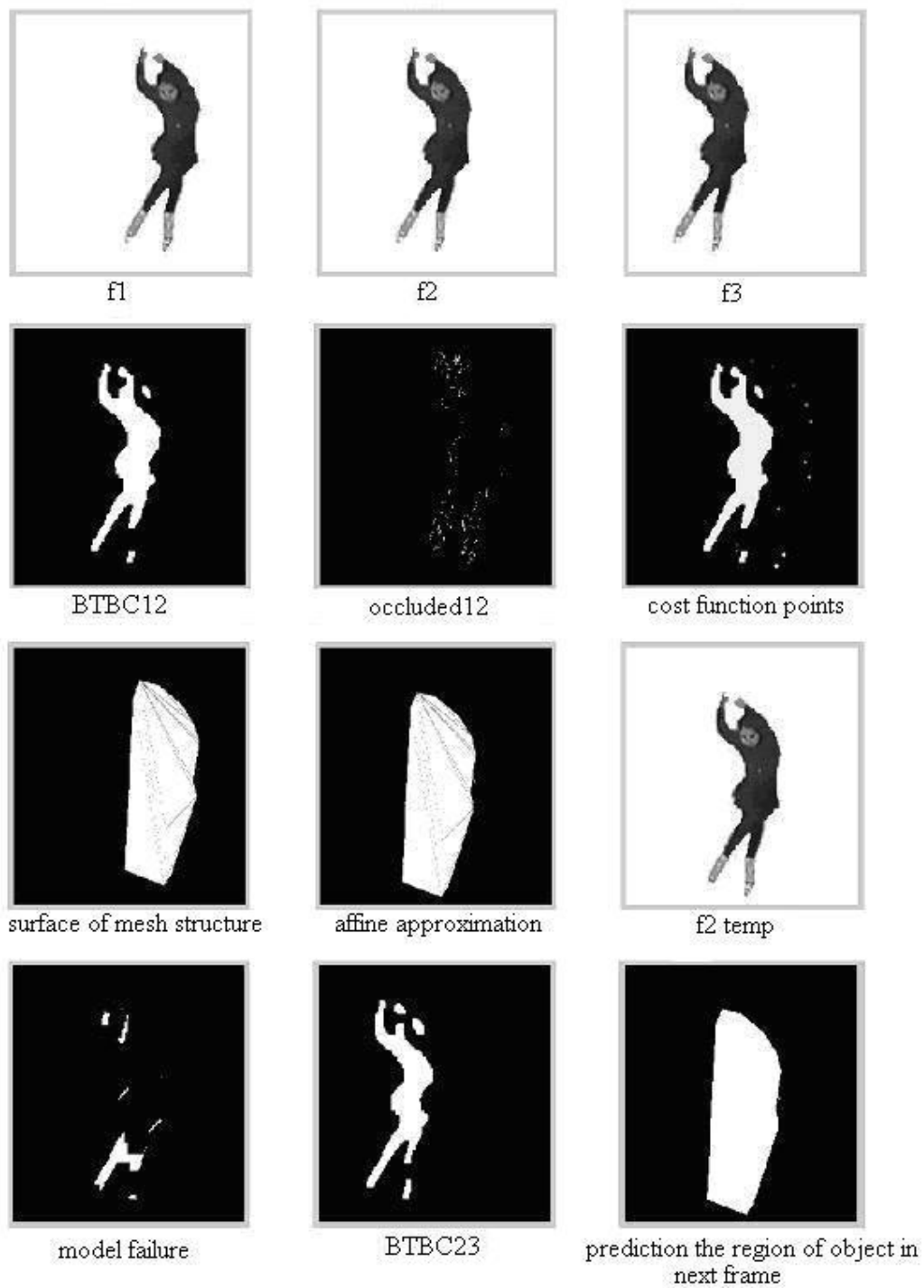
مراجع

[۱] Altunbasak, Y., Murat Tekalp, A., 1997. Occlusion-ADAPTIVE, Content-Based Mesh Design and Forward Tracking. IEEE, transaction on image processing, VOL. 6, NO. 9. PP.1270-1280.

[۲] Altunbasak, Y., Murat Tekalp, A., 1996. Occlusion-ADAPTIVE 2-D Mesh Tracking. IEEE. PP. 2108-2111.

[۳] Sayed, M., Badawy, W., 2004. A Novel Motion Estimation Method For Mesh-Based Video Motion Tracking. IEEE. PP. 337-340.

[۴] Beek, P., Murat Tekalp, A., 1993. Hierarchical 2-D Mesh Representation, Tracking, and Compression for Object-Based



شکل ۱۱: نتایج الگوریتم ردیابی مش و فقی