

رمزگذاری فریم تکیه گاه در کدگذاری تصویری

On the Encoding of the Anchor Frame in Video Coding

مصطفی ترابی کچوسنگی

چکیده:

در این مقاله، تعداد بیت های لازم برای رمزگذاری فریم تکیه گاه در کدگذاری تصویری با نرخ بیت پایین به منظور افزایش کیفیت فریم های بعدی و متعاقب که می خواهند رمزگذاری شوند، را مشخص می کنیم. این مساله در سیستم ارتباطات تصویری نوری از اهمیت ویژه ای برخوردار است. ما برای انتقال فریم تکیه گاه از یک شیوه پیش رونده استفاده می کنیم. دو شیوه را ایجاد کردیم تا تعداد بیت های بهینه ای که برای اولین فریم در اجرای ارتباطات تصویری Online در نظر گرفته می شوند را تعیین می کنیم.

۱- مقدمه:

در اغلب روش های امروزی برای کدگذاری تصویری با نرخ بیت پایین به تاثیر تعداد بیت های در نظر گرفته شده برای اولین فریم (تکیه گاه درونی) سکانس تصویر بر کیفیت کلی سکانس ترمیم شده، توجه نمی شود. برای فریم تکیه گاه از QP ^۱ (پارامتر کمی سازی) ثابت یا تعداد ثابتی از تعداد بیت ها استفاده می شود. با این وجود، این شیوه قطعا بهینه نمی باشد. با فرض این که ما از یک کانال با نرخ بیت ثابت استفاده می کنیم، تعداد بیت های به کار برده شده برای فریم دیگر متناظر با تاخیر زمانی مشخص است. به طور مثال، اگر نرخ بیت کانال، R بیت در ثانیه باشد و برای فریم تکیه گاه از τ بیت استفاده کنیم، انتقال آن فریم، زمانی برابر با " τ / R " ثانیه را صرف خواهد کرد. از این رو می توانیم بعد از گذشت این زمان (یعنی τ / R ثانیه)، فریم بعدی را کدگذاری کنیم. حین انتقال فریم تکیه گاه، تمام فریم های موجود برای کدگذار دور ریخته می شوند. فریم بعدی قرار است کدگذاری شود قبل از زمان انتقال یا دقیقا در زمان انتقال فریم تکیه گاه رمزگذاری شده، رسیده است.

به طور واضح برای انتقال فریم تکیه گاه باید یک زمان بهینه در نظر گرفته شود. البته، این زمان با زمان تمام سکانس های تصویری یکسان نیست. اگر بیت های زیادی را برای فریم درونی به کار ببریم به کیفیت بهتری از فریم درونی

^۱ Quantization Parameter

ترمیم شده دست پیدا خواهیم کرد. در واقع، کیفیت بهتر فریم درونی ترمیم شده منجر به کیفیت بهتر فریم میانی ترمیم شده بعدی می شود. اگر تاخیر زمانی به آن بزرگی نباشد باعث کمتر شدن رابطه و همبستگی میان دو فریم خواهد شد. بنابراین زمان به کار برده شده برای فریم درونی به اندازه کافی باید بزرگ باشد تا فریم ترمیم شده ای با کیفیت خوب حاصل شود، این زمان خیلی مهم نباید زیاد باشد زیرا اولین فریم و فریم بعدی که می خواهند کدگذاری شوند از هم گسسته خواهند شد.

همان طوری که پیش تر عنوان شد علاقه مندی ما بیشتر به کدگذاری تصویری با نرخ بیت پایین است. بنابراین فرض می کنیم که فقط یک فریم درونی یعنی اولین فریم فرکانس، ارسال می شود و تمام فریم های بعدی با استفاده از پیش بینی (فریم های میانی) کدگذاری می شوند. ما توجه و کار خود را معطوف استاندارد تراکم تصویری H.263 می کنیم. زیرا استاندارد بالفعل برای کدگذاری تصویری با نرخ بیت پایین می باشد.

برای تصمیم گیری در مورد اینکه چه زمانی باید کدگذاری فریم درونی را متوقف کنیم و چه زمانی باید کدگذاری فریم میانی بعدی را شروع کنیم، باید از یک سری اطلاعات و داده های مربوط به سکانس تصویری برخوردار باشیم. بنابراین بعد از آغاز کدگذاری فریم درونی و در دسترس قرار گرفتن فریم های بیشتر برای کدگذار، بلافاصله باید این تصمیم را اتخاذ کرد. این امر موجب می شود تا ما کدگذار H.263 را تغییر داده و اصلاح کنیم و برای کدگذاری فریم درونی از یک شیوه پیش رونده استفاده کنیم به طوری که بتوانیم در هر زمانی که می خواهیم کدگذاری آن را متوقف کنیم. شیوه ای که ما در این جا مورد استفاده قرار می دهیم تحت عنوان شیوه EZW¹ (موجک- موج کوچک- درخت صفر ریشه دار) نامیده می شود.

در این جا، ما فرض می کنیم که کانال ما یک کانال بدون خطا با نرخ بیت ثابت می باشد. بحث ما برای این مدل قابل اجرا است. مدل های کانال دیگری نیز وجود دارند که برای انتقال تصویر مورد استفاده قرار می گیرند، به طور مثال مدل های کانال با نرخ بیت متغیر که برای شبکه های ATM² (حالت انتقال غیر همزمان)، مناسب می باشند.

۲- شیوه EZW (موجک درخت صفر ریشه دار)

همان طوری که پیش تر عنوان شد، برای رمزگذاری فریم به یک شیوه پیش رونده نیاز داریم تا بتوانیم در هر زمانی که بخواهیم عمل انتقال را متوقف کنیم. این در حالی است که رمزگذار با استفاده از بیت های از قبل منتقل شده می تواند فریم تکیه گاه را رمزگذاری کند.

کدگذاری ریشه دار، مفهومی است که با شمایل های دقیق و متناهی دوتایی اعداد حقیقی مشابه است. تمام اعداد حقیقی را می توان با رشته ای از اعداد دوتایی (بیت های دوتایی) نشان داد. در نشان دادن هر بیتی که می خواهیم به

¹ Embedded Zerotree Wavelet
² Asynchronous Transfer Mode

سمت راست عدد دوتایی اضافه کنیم باید دقت بیشتری صورت گیرد. با این وجود، عمل رمزگذاری را در هر زمانی که بخواهیم می توانیم متوقف کنیم و با استفاده از تعداد بیت های مشخص نیز می توانیم عدد حقیقی را به بهترین نحو نشان دهیم. به همین ترتیب کدگذار، درخت صفر ریشه دار را در هر زمان می تواند متوقف کند و جریان بیت را نیز می تواند اندازه گیری کند. شیوه EZT انتقال موجک در تصویر را انجام می دهد. از این رو ضرایب انتقال موجک به تدریج رمزگذاری می شوند.

۳- بیان مساله:

همان طوری که در بخش مقدمه عنوان شد، ما می خواهیم یک نقطه زمانی پیدا کنیم که در آن زمان، باید رمزگذاری فریم تکیه گاه را متوقف کنیم. در بخش زیر علایمی را نشان می دهیم که در بقیه این مقاله از آن ها استفاده خواهیم کرد.

فرض می کنیم که می خواهیم یک سکانس تصویری را با نرخ فریم مشخصی در زمان حقیقی کدگذاری کنیم. این سکانس در یک کانال با نرخ ثابت منتقل می شود. "r" تعداد بیت هایی است که برای اولین فریم به کار می بریم، SNR₁ نسبت پیک سیگنال به (PSNR) noise¹ اولین فریم ترمیم شده می باشد و SNR₂ نیز PSNR فریم ترمیم شده بعدی است. SNR پیک تصویر ترمیم شده هشت بیتی (i, j) x به صورت زیر است:

$$PSNR = 10 \log \frac{255^2}{FD}, \quad (1)$$

$$FD = \frac{1}{N \cdot M} \sum_{i=1}^M \sum_{j=1}^N [x(i, j) - \hat{x}(i, j)]^2, \quad (2)$$

M و N ابعاد تصویر و (i, j) x تصویر اصلی است.

هدف ما حداکثرسازی SNR₂ می باشد به طوری که زمان تاخیر قابل قبول باشد و تعداد بیت های یکسانی برای فریم بعدی (دومین فریم رمزگذاری شده) در تمام زمان ها به کار برده شود تا مقایسه و تشابه خوبی صورت گیرد. بعد از دومین فریم رمزگذاری شده، ما برای رسیدن به نرخ بیت ثابت از کنترلر نرخ خود استفاده می کنیم. ما انتظار داریم که فاصله زمانی و به دنبال آن همبستگی میان بقیه فریم ها تقریباً شبیه هم باشند. بنابراین ما هم انتظار داریم که کیفیت خوب دومین فریم رمزگذاری شده بر فریم های بعدی تاثیر خواهد داشت، حداقل در یک اجرای کوتاه. به همین دلیل است که حداکثر سازی SNR₂ را به عنوان هدف خود انتخاب کنیم.

¹ Peak Signal- to-Noise Ratio

راه حلی که ما در نظر داریم بدین گونه است، این که SNR_1 و SNR_2 را به عنوان سری های زمانی با شاخص تعداد فریم های منبع در نظر بگیریم که این فریم ها متناظر با بیت های اولین فریم است. برای مثال، اگر نرخ فریم به تعداد ۳۰ فریم در ثانیه باشد و نیمی از این فریم ها را به دومین فریم برای کدگذاری اولین فریم اختصاص دهیم بنابراین فریم بعدی که می خواهد کدگذاری شود فریم شماره ۱۵ خواهد بود. از این رو PSNR فریم اول و فریم بعدی تحت عنوان $SNR_1[15]$ و $SNR_2[15]$ نامیده می شود.

ما فرض می کنیم که $SNR_2[n]$ یک ماکزیمم خواهد داشت یا این که حداقل یک ماکزیممی که مدّ نظرمان است را دارد و دلیل آن محدودیت های دیگر ما مثل تاخیر زمانی قابل قبول است. در بخش زیر از دو راه حل مختلف برای این مساله استفاده می کنیم که بستگی به این دارند که ما از چه داده هایی استفاده می کنیم و هنگامی که باید رمزگذاری فریم تکیه گاه را متوقف کنیم از چه داده های موجودی باید استفاده کنیم.

۴- نمونه شماره ۱: مقادیر قبلی $SNR_1[n]$ و $SNR_2[n]$ مقادیر معلومی هستند.

در این جا فرض می کنیم که مقادیر $SNR_1[n]$ و $SNR_2[n]$ که با شاخص n (شاخصی که متناظر با نقطه ای است که ما در آن قرار داریم) متناظر می باشند را می دانیم. همچنین همان طوری که قبلا عنوان شد ما فرض می کنیم $SNR_2[n]$ یک ماکزیمم دارد. از این رو، مساله ما ماکزیمم $SNR_2[n]$ را پیدا می کند. نتایج آزمایشگاهی نشان می دهند که $SNR_2[n]$ تقریباً همیشه به طور یکنواخت تا ماکزیمم آن افزایش می کند. از این رو می توانیم به انتظار بنشینیم تا $SNR_2[n]$ شروع به کاهش کند و در آن نقطه متوقف شود. به طور واضح ما به ماکزیمم $SNR_2[n]$ نخواهیم رسید، زیرا کمی کاهش پیدا خواهد کرد. شاخص n که ما در این شیوه به دست می آوریم همان " $n + 1$ بهینه" می باشد. عملاً ما کیفیت زیادی را در $SNR_2[n]$ از دست نمی دهیم زیرا کاهش مقدار آن کم است.

نقطه ضعف این شیوه این است که ما باید مقدار $SNR_2[n]$ برای هر n را محاسبه کنیم. این بدان معنی است که ما باید عملیات های جبران حرکت، تقییر کلی بلوک و ترمیم فریم دوم برای هر مقدار n را انجام دهیم.

۵- نمونه شماره ۲: مقادیر قبلی $SNR_2[n]$ معلوم نیست.

فرض کنید که مقدار $SNR_2[n]$ در زمان حقیقی را نمی توانیم محاسبه کنیم ولی در عوض $SNR_1[n]$ را داریم. ارزیابی $SNR_1[n]$ نیازمند محاسبه خیلی کمتری است بنابراین در واقع این طور فرض می کنیم که ما $SNR_1[n]$ را داریم ولی $SNR_2[n]$ را نداریم. همان طوری که پیش تر عنوان شد $SNR_2[n]$ تابعی از $SNR_1[n]$ و همبستگی

میان فریم اول و فریم n می باشد. مقیاس کمی برای همبستگی می تواند اختلاف فریم (FD) یا بهتر بگوییم اختلاف فریم جابجا شده (DFD) می باشد. FD میان تصاویر $x(i, j)$ و $y(i, j)$ به صورت زیر تعیین می شود:

$$FD = \frac{1}{N \cdot M} \sum_{i=1}^M \sum_{j=1}^N [x(i, j) - y(i, j)]^2, \quad (3)$$

در حالی که DFD میان تصاویر $x(i, j)$ و $y(i, j)$ به صورت زیر تعیین می شود:

$$DFD = \frac{1}{N \cdot M} \sum_{i=1}^M \sum_{j=1}^N [\tilde{x}(i, j) - y(i, j)]^2, \quad (4)$$

جایی که $\tilde{x}(i, j)$ همان تصویر اصلی جابجا شده بعد از انجام جبران حرکت می باشد.

برای این که FD و DFD در واحدهای مشابه (dB) مثل $SNR_1[n]$ و $SNR_2[n]$ قرار گیرد ما از کمیتهای زیر استفاده می کنیم:

$$FD_{dB} = 10 \log \frac{255^2}{FD}, \quad (5)$$

و

$$DFD_{dB} = 10 \log \frac{255^2}{DFD}. \quad (6)$$

ما فرض می کنیم که سیستمی داریم با $SNR_1[n]$ و همبستگی مثل ورودی ها و $SNR_2[n]$ مثل خروجی. اسن سیستم به صورت غیر خطی است ولی از روی حدس می توان انتظار آن را داشت که این سیستم بدون حافظه است. این مساله می تواند این سیستم غیر خطی و بدون حافظه را شناسایی کند.

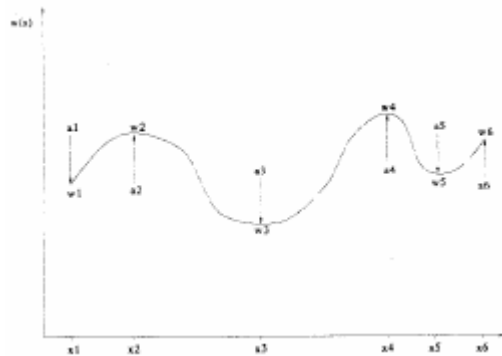
با فرض این که از DFD به عنوان مقیاس همبستگی استفاده می کنیم این برابر با تخمین تابع است:

$$SNR_2[n] = f(SNR_1[n], DFD[n]). \quad (7)$$

این رویکرد درصدد یافتن برخی نقاط داده این تابع با استفاده از داده های آزمایشی است و سپس از درون یابی برای تخمین مقدار $SNR_2[n]$ با معلوم بودن مقادیر $SNR_1[n]$ و $DFD[n]$ استفاده می کند. شیوه درون یابی که ما به کار می بریم "درون یابی Biharmonic Spline" را در یک بعد، دو بعد یا بیشتر توصیف می کنیم.

شکل ۱. تابع $w(x)$ با استفاده از نیروهای نقطه ای به یک میله لاستیکی پیدا شده است.

^۱ Frame Difference
^۲ Displaced Frame Difference



شکل ۱

A: درون یابی Biharmonic Spline در یک بعد:

امیدواریم تابع biharmonic ای را پیدا می کنیم که از نقاط داده های N عبور کنند. طراحان قرن ۱۹ با در نظر گرفتن وزنه هایی به میله لاستیکی و قرار دادن این وزنه ها توانستند این مساله را حل کنند به طوری که این میله از نقاط داده

عبور می کردند. نیروهای تحمیل شده از سوی هر وزنه بر این میله باعث خم شدن آن می شوند. برای $\phi(x)$ جابجایی های کوچک، این میله به غیر از وزنه ها دارای مشتق چهارم صفر می باشد (به شکل ۱ مراجعه کنید). تابع Green نیروی نقطه ای برای میله با معادله زیر صدق می کند:

$$\frac{d^4 \phi}{dx^4} = \delta(x), \quad (8)$$

جایی که $\delta(x)$ بیانگر تابع دلتا است.

جواب و راه حل معادله ۸ به صورت زیر است:

$$\phi(x) = |x|^3. \quad (9)$$

هنگامی که از تابع Green $w(x)$ برای درون گیری نقاط داده های N یعنی w_i مستقر در x_i استفاده می شود، مساله

به صورت جواب زیر در می آید:

$$\frac{d^4 w}{dx^4} = \sum_{j=1}^N \alpha_j \delta(x - x_j),$$

with $w(x_i) = w_i.$ (10)

جواب و راه حل معادله ۱۰، ترکیب خطی توابع Green نیروی نقطه ای متمرکز در هر نقطه داده است که به صورت زیر می باشد:

$$w(x) = \sum_{j=1}^N \alpha_j |x - x_j|^3. \quad (11)$$

طول هر یک از نیروهای نقطه ای یعنی α_j را می توان با حل سیستم معادلات خطی زیر به دست آورد:

$$w_i = \sum_{j=1}^N \alpha_j |x_i - x_j|^3. \quad (12)$$

هنگامی که α_j ها مشخص می شوند، تابع biharmonic $w(x)$ را می توان در هر نقطه ای با استفاده از معادله ۱۱ ارزیابی کرد.

A: درون یابی Biharmonic Spline در دو بعد یا ابعاد بیشتر:

مشتقات تکنیک دو بعدی یا بیشتر شبیه مشتقات یک بعدی است. برای نقاط داده های N در ابعاد m ، مساله به صورت زیر است:

$$\begin{aligned} \nabla^4 w(\mathbf{x}) &= \sum_{j=1}^N \alpha_j \delta(\mathbf{x} - \mathbf{x}_j), \\ \text{with } w(\mathbf{x}_i) &= w_i, \end{aligned} \quad (13)$$

جایی که اپراتور biharmonic ∇^4 و \mathbf{x} وضعیتی در فضای m بعدی است. راه حل کلی یا جواب کلی به صورت زیر است:

$$w(\mathbf{x}) = \sum_{j=1}^N \alpha_j \phi_m(\mathbf{x} - \mathbf{x}_j). \quad (14)$$

با حل سیستم معادلات خطی به مقدار α_j پی خواهیم برد.

$$w_i = \sum_{j=1}^N \alpha_j \phi_m(\mathbf{x}_i - \mathbf{x}_j). \quad (15)$$

هنگامی که α_j ها مشخص شدند تابع biharmonic $w(x)$ را می توان در هر نقطه ای با استفاده از معادله ۱۴ ارزیابی کرد.

$$\phi_m$$

توابع biharmonic Green یعنی ϕ_m برای هر بعد در جدول ۱ نشان داده شده اند. همان طوری که از این جدول مشهود است در دو بعد که ما به آن علاقه داریم، تابع Green برابر با مقدار زیر است:

$$\phi_2(\mathbf{x}) = |\mathbf{x}|^2 (\ln |\mathbf{x}| - 1). \quad (16)$$

Num. of Dimensions m	Green Function $\phi_m(x)$
1	$ x ^3$
2	$ x ^2(\ln x - 1)$
3	$ x $
4	$\ln x $
5	$ x ^{-1}$
6	$ x ^{-2}$
m	$ x ^{4-m}$

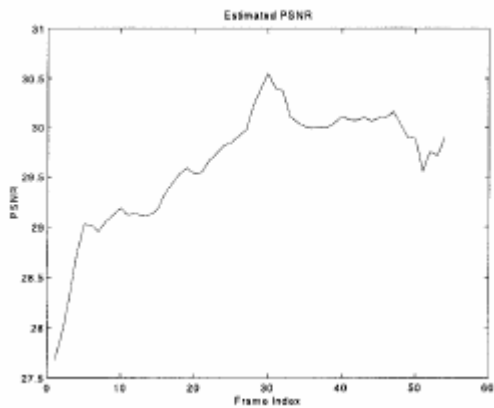
جدول ۱: جدول توابع BIHARMONIC GREEN

C: نتایج آزمایشگاهی:

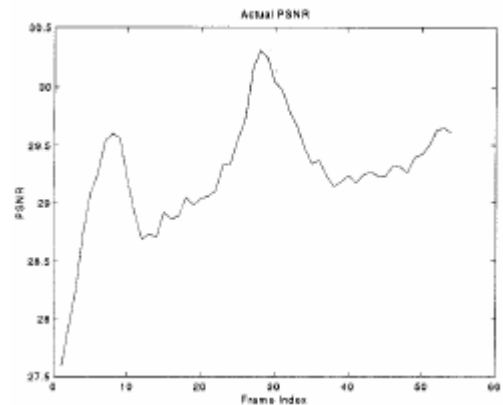
در مثال زیر، با استفاده از مقادیر واقعی $SNR_1[n]$ ، DFD_{dB} و سکانس "مادر و دختر" می توانیم مقادیر SNR_2 را تخمین بزنیم. از سکانس های دیگر نیز استفاده می شود و هنگامی که مقدار حرکت با مقدار سکانس "مادر و دختر" قابل مقایسه باشد، نتایج حاصل از آن ها نیز مشابه هم هستند. در این جا باید تاکید شود که ما فقط به ماکزیمم منطقه ای $SNR_2[n]$ علاقه مند هستیم نه به مقادیر واقعی شان. عملاً گاهی اوقات "سطح DC" از $SNR_2[n]$ تخمین زده شده با سطح اصلی متفاوت است ولی در اغلب موارد شاخص n که متناظر با ماکزیمم منطقه ای است با صحت و درستی خیلی خوبی تخمین زده می شود. چند نمونه اول $SNR_1[n]$ و $SNR_2[n]$ نادرست هستند زیرا شیوه ی EZW نمی تواند تصاویری با کیفیت قابل قبول و نرخ بیت خیلی پایین را ایجاد کنید. بنابراین در شبیه سازی زیر، پنج نمونه اول تمام سکانس ها کنار گذاشته اند.

شکل ۲ مقادیر واقعی $SNR_2[n]$ را نشان می دهد و شکل ۳ مقادیر تخمین زده شده ی $SNR_2[n]$ را نشان می دهد. $SNR_2[n]$ از شروع سکانس "مادر و دختر" در فریم "۰" (مقادیر پنج مورد اول مورد استفاده قرار نگرفته اند)، محاسبه می شود. همان طوری که می توان مشاهده کرد ماکزیمم حقیقی در شاخص ۸ می باشد در حالی که ماکزیمم $SNR_2[n]$ تخمین زده شده، ۵ است. شکل ۴ و شکل ۵ مقدار $SNR_2[n]$ حقیقی و تخمین زده شده را نشان می دهند، این هنگامی است که ما از فریم ۵۰ سکانس "مادر و دختر" شروع می کنیم. در این مورد ماکزیمم حقیقی در شاخص ۱۳ می باشد و ماکزیمم تخمین زده شده، ۹ است.

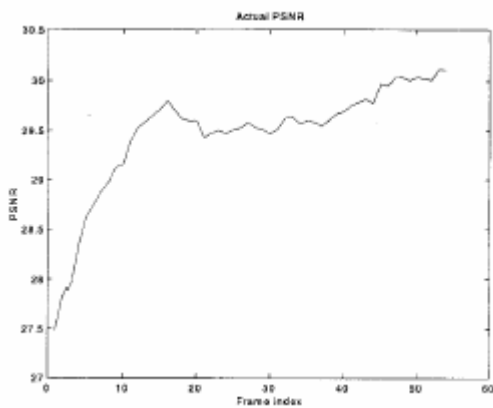
به طور واضح، از این مثال ها و نمونه ها می بینیم که اولین ماکزیمم $SNR_2[n]$ را به طور کاملاً درست می توانیم در بسیاری از موارد با استفاده از شیوه در نظر گرفته شده، تخمین زد.



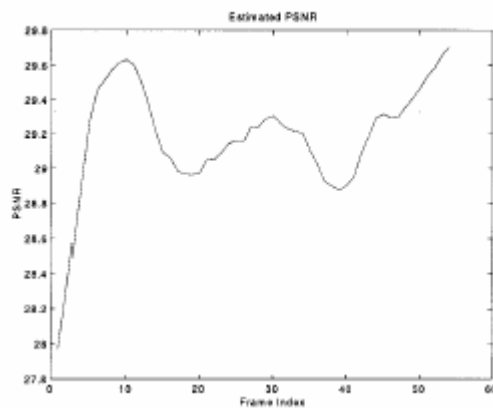
شکل ۳



شکل ۲



شکل ۵



شکل ۴

نتیجه گیری:

انگیزه این کار انتظار حسی بود که بیت ها می بایستی به صورت بهینه به فریم نکیه گاه تخصیص داده شوند، این فریم بهترین کیفیت را برای فریم بعدی ارائه می دهد. این تخصیص بیت بستگی به سکانس تصویر دارد و باید به صورت Online تعیین شود، از این رو باید از یک شیوه پیش رونده (شیوه تدریجی) برای انتقال فریم تکیه گاه استفاده شود. ما از حداکثرسازی SNR_2 به عنوان هدف خود استفاده کردیم یعنی هدف خود را به حداکثر رساندن SNR_2 تعیین کردیم زیرا می خواهیم بعد از فریم دوم به نرخ فریم ثابت برسیم و این فرض معقول و منطقی است که همبستگی میان هر دو فریم رمزگذاری شده بعدی تقریباً یکسان است. این صحیح خواهد بود مگر این که در تصویر یک تغییر چشمگیر صورت

گیرد که نتوان آن را پیش بینی کرد. از این رو کیفیت بهتر فریم دوم رمزگذاری شده معمولا به معنی کیفیت بهتر فریم های باقیمانده است، حداقل در اجرای کوتاه مدت یا تا زمانی که تغییر چشمگیری در صحنه ایجاد شود. نتایج آزمایشگاهی نشان می دهند که به غیر از مواردی که در آن ها حرکت خیلی کمی وجود دارد، اولین ماکزیمم را می توان در SNR_2 مشاهده کرد. بنابراین مشخص است که اگر برای فریم تکیه گاهی که در SNR_2 پیک دارد، یک سری بیت را تخصیص بدهیم ما می توانیم کیفیت کلی بهتری را برای سکانس تصویر (مطابق با بحث قبل) انتظار داشته باشیم.

اگر پیچیدگی محاسباتی را در نظر نگیریم، یافتن اولین ماکزیمم در SNR_2 کاری آسان است. از نتایج آزمایشگاهی می دانیم که $SNR_2 [n]$ تقریبا همیشه تا اولین ماکزیمم افزایش پیدا می کند. بنابراین اگر $SNR_2 [n]$ را به صورت Online بتوانیم محاسبه کنیم، با تعیین مکانی که در آن $SNR_2 [n]$ شروع به کاهش می کند می توانیم ماکزیمم را مستقر کنیم. بنابراین پیچیدگی محاسباتی در این مورد خیلی زیاد است زیرا لازم است تا هر $1/30$ ام ثانیه در مورد نرخ فریم منبع 30 فریم در ثانیه، دومین فریم را رمزگذاری و ترمیم کنیم.

کارهای دیگر در این حیطه می توانند مستلزم توسعه یک الگوریتم پیچیده تری برای مکان ماکزیمم باشد که قادر به شناسایی ماکزیمم منطقه ای خیلی کوچکی خواهد بود که در آن SNR_2 مقدا کمی کاهش پیدا می کند و سپس سریعا تا ماکزیمم مشخصی افزایش پیدا می کند. الگوریتم جدید باید قادر به نادیده گرفتن چنین ماکزیمم منطقه ای کوچکی باشد. اگر فرض کنیم که نتوانیم $SNR_2 [n]$ را به صورت Online محاسبه کنیم، می توانیم این طور فرض کنیم که $SNR_2 [n]$ تابعی از $SNR_1 [n]$ است و همبستگی میان دو فریم به وسیله انرژی DFD اندازه گیری می شود. در این کار، ما با استفاده از درون یابی داده های آزمایشگاهی این تابع را تخمین می زنیم. پیچیدگی محاسباتی این مورد خیلی کمتر است زیرا برای یافتن DFD باید جبران حرکت را انجام دهیم ولی مجبور نیستیم تمام مراحل دیگر مورد نیاز برای یافتن فریم دوم ترمیم شده و محاسبه PSNR آن را انجام دهیم.

نتایج آزمایشگاهی نشان داده اند که با استفاده از شیوه بالا می توانیم به طور رضایت بخش ماکزیمم در $SNR_2 [n]$ را تخمین بزنیم. در اغلب موارد، موقعیت تخمین زده شده ی ماکزیمم، خیلی نزدیک موقعیت حقیقی است. بنابراین، در این مقاله دریافتیم که ارسال تدریجی فریم تکیه گاه و استفاده از شیوه های تعیین بهترین نقطه توقف، ارزشمند و مفید است. شیوه هایی که در این مقاله شرح داده شده اند، این نقطه توقف را به طور کاملا صحیحی می توانند تخمین بزنند.