Subject: Re: RMS error
Posted by David Fanning on Tue, 11 May 2004 21:14:39 GMT
View Forum Message <> Reply to Message

Julio writes:

> I'd like to calculate Root Mean Square error using a base image and a
> secondary image. How can I do that? Clues are welcome!

rms_error = Sqrt(Total((img_1 - img2)^2)/N_Elements(img_1))

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: RMS error
Posted by Rick Towler on Tue, 11 May 2004 22:34:43 GMT
View Forum Message <> Reply to Message

"David Fanning" wrote ...
> Julio writes:
>
>> I'd like to calculate Root Mean Square error using a base image and a
>> secondary image. How can I do that? Clues are welcome!
>
> rms_error = Sqrt(Total((img_1 - img2)^2)/N_Elements(img_1))

A couple of questions:

1) Shouldn't you first cast img_1 and img_2 to something other than byte?
Subtracting byte arrays will result in points that wrap:

IDL> print, 213B-215B
 254

When you really want:

IDL> print, 213s-215s
      -2

2) Is this correct for true color images? Do you total the square of the

differences for each color plane or do you total the square of the sum of the differences for each plane? That is:

```
IDL> print, TOTAL((img_1 - img_2)^2)
     56825.0
```

or:

```
IDL> dr=reform(img_1[0,*,*] - img_2[0,*,*])
IDL> dg=reform(img_1[1,*,*] - img_2[1,*,*])
IDL> db=reform(img_1[2,*,*] - img_2[2,*,*])

IDL> print, total((dR + dG + dB)^2)
     170325.
```

Now that I think about it the first approach looks correct but since I am not an image analyst and I don't have even the most basic of references I thought I would ask.

-Rick

Subject: Re: RMS error
Posted by David Fanning on Tue, 11 May 2004 23:23:02 GMT
View Forum Message <> Reply to Message

Rick Towler writes:

> A couple of questions:
>
> 1) Shouldn't you first cast img_1 and img_2 to something other than byte?
> Subtracting byte arrays will result in points that wrap:
>
> IDL> print, 213B-215B
> 254
>
> When you really want:
>
> IDL> print, 213s-215s
> -2

Yes, of course. I assumed these were 16-bit images, because, well..., I just did. :-)

> 2) Is this correct for true color images?

Don't know. I've never done this myself. I was just up
getting a cup of tea when the message came in and I
happened to have my Gonzalez and Woods sitting at hand.
This is what I found when I looked RMS error up in the
index. It's what happens when you give a guy with a
compulsion to communicate a little bit of knowledge. :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: RMS error
Posted by Craig Markwardt on Wed, 12 May 2004 14:13:53 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> writes:

> Julio writes:
>
>> I'd like to calculate Root Mean Square error using a base image and a
>> secondary image. How can I do that? Clues are welcome!
>
> rms_error = Sqrt(Total((img_1 - img2)^2)/N_Elements(img_1))

Hey David, I'll pile on too. The problem I see with the above
expression is that there could be an offset between the two images,
which you are including in your TOTAL expression, and hence biasing
the rms value. How about the following instead?

rms_error = stddev(img_1 - img2)

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: RMS error
Posted by David Fanning on Wed, 12 May 2004 14:42:28 GMT
View Forum Message <> Reply to Message

Craig Markwardt writes:

> Hey David, I'll pile on too. The problem I see with the above
> expression is that there could be an offset between the two images,
> which you are including in your TOTAL expression, and hence biasing
> the rms value.

An offset!? We better take global warming into
account, too. :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: RMS error
Posted by Paul Van Delst[1] on Wed, 12 May 2004 14:56:47 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
> David Fanning <david@dfanning.com> writes:
>
>
>> Julio writes:
>>
>>
>>> I'd like to calculate Root Mean Square error using a base image and a
>>> secondary image. How can I do that? Clues are welcome!
>>
>> rms_error = Sqrt(Total((img_1 - img2)^2)/N_Elements(img_1))
>
>
> Hey David, I'll pile on too. The problem I see with the above
> expression is that there could be an offset between the two images,
> which you are including in your TOTAL expression, and hence biasing
> the rms value.

Errmm... isn't that intended? RMS errors include the bias. Std deviations don't. At any rate, that's how I use 'em. This is what I've got on my wall:

RMS = SQRT(TOTAL(Xtruth - Xpredicted)^2 / N)

and

StdDev = SQRT(TOTAL(Xpredicted - MEAN(Xpredicted))^2 / N)

I think it really requires the OP to state more precisely what it is he's looking for.
Blind use of statistical formula is usually a no-no.

> rms_error = stddev(img_1 - img2)

That works - same as the RMS defn above where Xtruth = img1-MEAN(img1) and Xpredicted = img2-MEAN(img2). But is that what the OP needs/wants?

paulv

Subject: Re: RMS error
Posted by James Kuyper on Wed, 12 May 2004 15:02:19 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
> David Fanning <david@dfanning.com> writes:
>
>
>> Julio writes:
>>
>>
>>> I'd like to calculate Root Mean Square error using a base image and a
>>> secondary image. How can I do that? Clues are welcome!
>>
>> rms_error = Sqrt(Total((img_1 - img2)^2)/N_Elements(img_1))
>
>
> Hey David, I'll pile on too. The problem I see with the above
> expression is that there could be an offset between the two images,
> which you are including in your TOTAL expression, and hence biasing
> the rms value. How about the following instead?

The Root Mean Square difference between two images is defined as the
square ROOT of the MEAN of the SQUAREd differences between the images.
If there's an offset between the two images, that offset is supposed to
be squared, and is supposed to contribute to the mean, and therefore to
an increase in the RMS error.

> rms_error = stddev(img_1 - img2)

That's a different statistic, also useful, but it's not the RMS error.

Note: if there's an offset difference, there might also be a scaling difference. Then the most appropriate statistic to use gets even more complicated.

Subject: Re: RMS error
Posted by Craig Markwardt on Wed, 12 May 2004 15:11:43 GMT
View Forum Message <> Reply to Message

Paul Van Delst <paul.vandelst@noaa.gov> writes:

> Craig Markwardt wrote:
>> David Fanning <david@dfanning.com> writes:
>>
>>
>>> Julio writes:
>>>
>>>
>>>> I'd like to calculate Root Mean Square error using a base image and a
>>>> secondary image. How can I do that? Clues are welcome!
>>>
>>> rms_error = Sqrt(Total((img_1 - img2)^2)/N_Elements(img_1))
>>
>>
>> Hey David, I'll pile on too. The problem I see with the above
>> expression is that there could be an offset between the two images,
>> which you are including in your TOTAL expression, and hence biasing
>> the rms value.
>
> Errmm... isn't that intended? RMS errors include the bias. Std deviations don't. At any
> rate, that's how I use 'em. This is what I've got on my wall:

It's a good question for the original poster!

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: RMS error
Posted by julio on Thu, 13 May 2004 13:22:57 GMT

Hello all...

I'd like to thank you for the discussion and the clues! However, I'd
like to know a way to extract from the image the coordinates to be put
in the RMS equation.
Thanks...
Julio

James Kuyper <kuyper@saicmodis.com> wrote in message
news:<40A23C7B.3070003@saicmodis.com>...
> Craig Markwardt wrote:
>> David Fanning <david@dfanning.com> writes:
>>
>>
>>> Julio writes:
>>>
>>>
>>>> I'd like to calculate Root Mean Square error using a base image and a
>>>> secondary image. How can I do that? Clues are welcome!
>>>
>>> rms_error = Sqrt(Total((img_1 - img2)^2)/N_Elements(img_1))
>>
>>
>> Hey David, I'll pile on too. The problem I see with the above
>> expression is that there could be an offset between the two images,
>> which you are including in your TOTAL expression, and hence biasing
>> the rms value. How about the following instead?
>
> The Root Mean Square difference between two images is defined as the
> square ROOT of the MEAN of the SQUAREd differences between the images.
> If there's an offset between the two images, that offset is supposed to
> be squared, and is supposed to contribute to the mean, and therefore to
> an increase in the RMS error.
>
>> rms_error = stddev(img_1 - img2)
>
> That's a different statistic, also useful, but it's not the RMS error.
>
> Note: if there's an offset difference, there might also be a scaling
> difference. Then the most appropriate statistic to use gets even more
> complicated.

Subject: Re: RMS error
Posted by David Fanning on Thu, 13 May 2004 13:36:50 GMT

Julio writes:

> I'd like to thank you for the discussion and the clues! However, I'd
> like to know a way to extract from the image the coordinates to be put
> in the RMS equation.

Sigh...

I'll let somebody else field this.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/